Sudan University of Science and Technology

College of Graduate Studies

Design of a Control Board for Clean Water Supply to Gland System in Sennar Power Station

تصميم لوحة تحكم لتزويد الماء النظيف لنظام Gland بمحطة توليد سنار

A thesis Submitted as partial fulfillment of the requirements for the Degree of Master of Mechatronics Engineer

By:

Mohamed Noraldin Abdalla

Supervisor:

Dr. Mohamed Elnour AbdAlla

December 2018

الآيـــــة

بسم الله الرحمن الرحيم

قــال تعــالــى :

﴿ وَلَقَدْ آتَيْنَا دَاوُودَ وَسُلَيْمَانَ عِلْمًا وَقَالَا الْحَمْدُ لِلَّهِ الَّذِي فَضَّلَنَا عَلَى كَثِيرٍ مِنْ عِبَادِهِ الْمُؤْمِنِينَ ﴾

صدق الله العظيم

سورة النمل – الآية (15)

DEDICATION

I dedicate this work to:

My parents

And

Small family

For their love, and great support towards my education.

ACKNOWLEDGEMENT

First and foremost, praises and thanks to Allah, over the years of my MSc. course work and research, I was very fortunate to enjoy the guidance, friendship, and support of many individuals who enriched my research and my life either through their supervision, suggestions, and technical discussion. Words cannot express my appreciation to my supervisors **Dr. Mohamed Elnour AbdAlla** . His guidance and support throughout the research period are key elements for the success of this work.

Acknowledgement is due to Sudan University of Science and Technology for giving me the opportunity of studying Mechatronis Engineering which has really broadened my horizon and opened new chances for me.

I also acknowledge my colleagues in Sennar Hydro Power Plant, for their encouragement and support.

Last but not least, I would like to express my sincere love and gratitude to my parents and my dear wife for their perpetual support and prayers.

ABSTRACT

Hydro Power Plants and Power Plants in general, consist of various components. The main shaft seal (gland) in a hydro-turbine is one of the key components of the turbine system. In order to maintain it, gland must operate in the ideal conditions of operation by injecting clean water in the gland to neutralize the pressure of the raw water from the river. The aim of this thesis is to design a physical model of the old gland system for the Kaplan turbine, the model developed to monitor and control the pressure of raw water from the river, the pressure of clean injected water and the water level in the gland sum continuously and permanently. An electronic control board (arduino) used with pressure and level sensor. Proteus 8 Professional used to evaluate the result from the simulation. Good results have been given from model and simulation.

المستخلص

تحتوى محطات التوليد المائى للكهرباء على عدة مكونات رئيسية، ويعتبر الجلند من الاجزاء المهمة في التوربينات المائية وللحفاظ عليه يجب ان يعمل في الظروف المثالية الخاصة بتشغيله وذلك بضخ ماء نظيف في الجلند لمعادلة ضغط الماء الخام من النهر. الهدف من هذه الأطروحة هو تصميم نموذج فيزيائي لنظام الجلند القديم لتوربينة كابلان، تم تطوير نموذج للمراقبة والتحكم في مستوي ضغط الماء الخام من النهر وضغط مضخات الجلند ومستوي الماء في حوض الجلند بصورة مستمرة ودائمة. باستخدام لوحة تحكم الكترونيه (اردينو) مع حساس ضغط وحساس مسافة. استخدم برنامج بروتس 8 الاحترافي لعمل نمذجة ومحاكاه للنموزج لاخذ النتائج. تم التوصل الي نتائج جيدة من النموذج والمحاكاه.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

ALU	Arithmetic Logic Unit
BJT	Bipolar Junction Transistor
Cm	Centimeter
DC	Direct Current
I	Current
ICSP	In Circuit Serial Programming
LCD	Liquid Crystal Display
US	Ultrasonic
PMDC	Permanent Magnet Direct Current
PPS	Pump Pressure Sensor
PWM	Pulse-Width Modulation
RWPS	Raw Water Pressure Sensor
RX	Receive
SPI	Serial Peripheral Interface
TX	Transmit
TTL	Transistor-Transistor Logic
USB	Universal Serial Bus
V	Voltage

CHAPTER ONE

INTRODUCTION

1.6 Background:

Sennar hydropower station is the oldest hydro-power station in Sudan, which was built in 1962; it located beside the Dam on the left bank of the Blue Nile, Two Kaplan turbines of 7.5 MW each.

The main shaft seal (gland) in a hydro-turbine is one of the key components of the turbine system. Ideally, the seal should seal completely but due the size of typical units such an arrangement is prohibitively expensive. The main shaft seal therefore functions not to eliminate leakage but more to control the leakage to an acceptable amount. The requirements on such seals include, but are not limited to, effective operation, long wears life, easy maintenance and low initial cost. The main shaft seal often operates under harsh working conditions. Two key challenges for the design are that the seal must provide high wear resistance against abrasives within the water and be able to operate at high rubbing velocities with low leakage. [1] A flow of clean water was injected to the seal to provide cooling, lubrication and to reduce wear of the sealing surfaces due to abrasive particles. Water should be filtered to ensure that large impurities are removed.

1.7 Problem statement:

The system of clean water supply to shaft seals in sennar hydropower station has completed its design life time, so it’s removes from the turbine system because of lack of spare parts. And these days the shaft seal is working by the carbon segment facing the raw water, and that lead to quick wearing in carbon segment, of course more leakage from shaft seal and increase maintenance downtime.

1.8 Objectives:

The aim of this project is to design a control board or physical model of the old clean water supply to gland system" in sennar hydropower station. And evaluate the results from the hardware model and simulation. For a possibility of restoring it in the turbine in a modern way to set these goals:

- To increase the life time of shaft seal (carbon segment) by working in clean water.
- To reduce the number of submersible pump that used for excess leakage water.
- To reduce the working time of main discharge pump.
- To reduce the consumption of electricity in the plant.

1.9 Methodology:

In this project, Arduino board use with three input Signals and one output signal. Two of input signals are employed to read water pressures, one from pressure sensor before shaft seal (raw water), and the other from pressure sensor of clean injected water (pump pressure), and the third input signal from ultrasonic distance sensor to read the level of water in gland sum. These pressure input signals used to generate output signal to control the pressure of clean injected water by controlling the speed of the motor pump instead of pressure control valve, and equalized the pump pressure a bit higher than pressure of raw water. And other output signal that used as alarm in case of the level of water in gland sum was high.

1.10Research layouts:

This thesis is divided into five chapters:

Chapter one which is a general introduction to the turbine shaft seal and clean water supply to shaft seal and discussed the purpose of this thesis.

Chapter two contains a detailed description to old system and the components that used in this project.

Chapter three is detailed of Installation the model, electronic circuit used in details and implementation procedure of project micro controller (ardiuno program)

Chapter four is dedicated for the observations and results.

Chapter five is conclusion and summery of the work, also recommendations are proposed in this chapter.

CHAPTER TWO

COMPONENTS DESCRIPTION

2.1 Turbine shaft seale:

As show in figure (2.1), there are generally two main types of seals in widespread service, differentiated by the orientation of the sealing face to the shaft axis, these are radial and axial seals. For the larger shaft diameters (over 1000 mm), axial type seals are receiving more preference in recent years. [1]

Two versions of the basic “axial” face seal concept have been used: hydrostatic, which requires clean water injection into the sealing interface, and hydrodynamic, which generators its own interface liquid film by using the fluid being sealed.

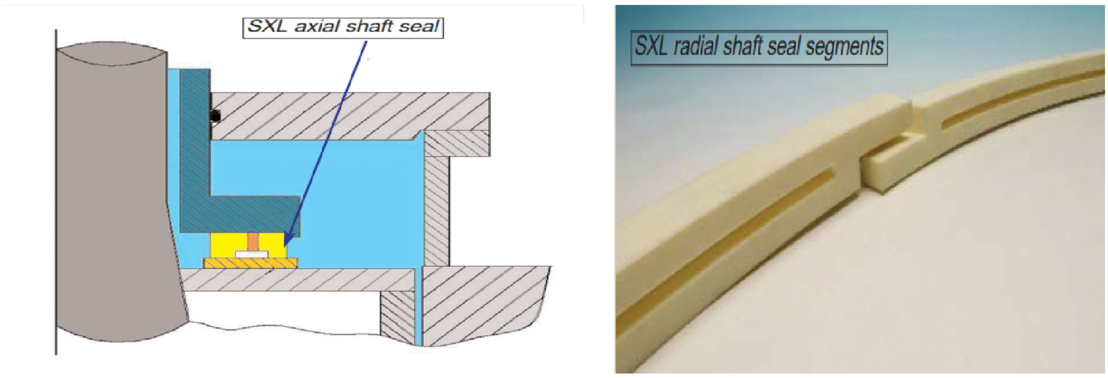

Figure (2.1): radial and axial seals.

The use of hydrostatic seals is especially advantageous when the following special operation requirements exist:

- Lowest possible face wear rate.
- Contaminated operating water, particularly when the contaminants are abrasive solids.
- Synchronous condenser operation (with runner operating in air).

The packing-box type of main shaft-seal is the simplest configuration of the mentioned shat seals. The sealing element is generally a series of square woven packing compressed by a packing gland to provide sufficient packing pressure on the main shaft to provide controlled leakage through the packing box. It should be pointed out that packing box seals require a small amount of leakage to cool the seal properly. [2]

The turbine shaft is sealed with carbon rings. These are composed of several segments with free movement between the joints. To prevent these being pushed together in Operation, each segment is fixed to the casing wall. The contact pressure of the rings should lie between 1.5 ... 2.0 N/cm2. [3]

Turbine shaft seal (gland) in sennar power station is consist of ring composed of a six stages of carbon segments held together by a peripheral close-coiled tension or garter spring. The ends of these segments are chamfered to present sliding surfaces to tapered wedge pieces. Glands of this type are self-adjusting, therefore segments moves radial inwards under the force of the spring as wear takes place. Figure (2.2) show shaft seal components.

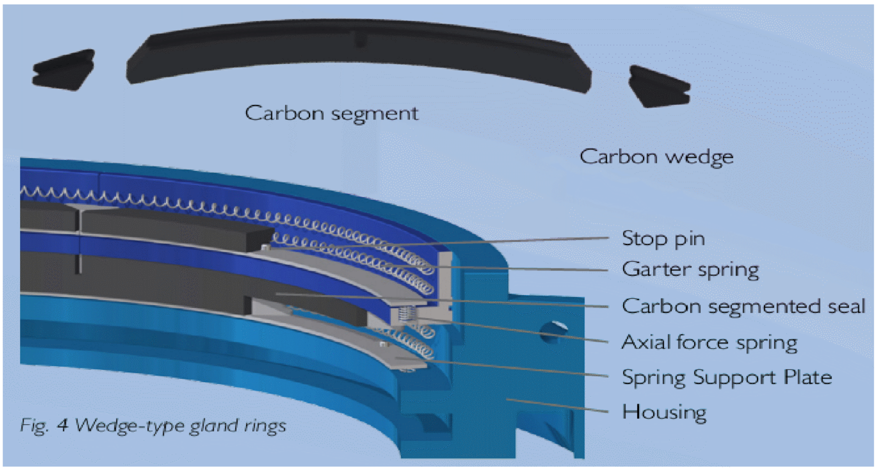

Figure (2.2): internal shape of shaft seal

Excessive Water Leakage from the turbine shaft seal (Gland), caused significant losses in efficiency and creates unnecessary damage and cost. This damage can come in the form of flooding of the turbine guide bearing or corrosion of the surrounding structure. All seals must leak to survive, In fact, acceptable leakage is only the amount required to sufficiently cool and lubricate the sealing faces.

So, a flow of clean water was injected to the seal to provide cooling, lubrication and to reduce wear of the sealing surfaces due to abrasive particles. Water should be filtered to ensure that large impurities are removed, in order to prevent abrasives particles entering from under the seal. The clean water supply must be at a higher pressure than the water pressure under the seal. It is recommended to install an automatic differential pressure system with the water supply, maintaining a constant pressure difference above that of the pressure under the seal. Figure (2.3) shows the arrangement of carbon segment seal.

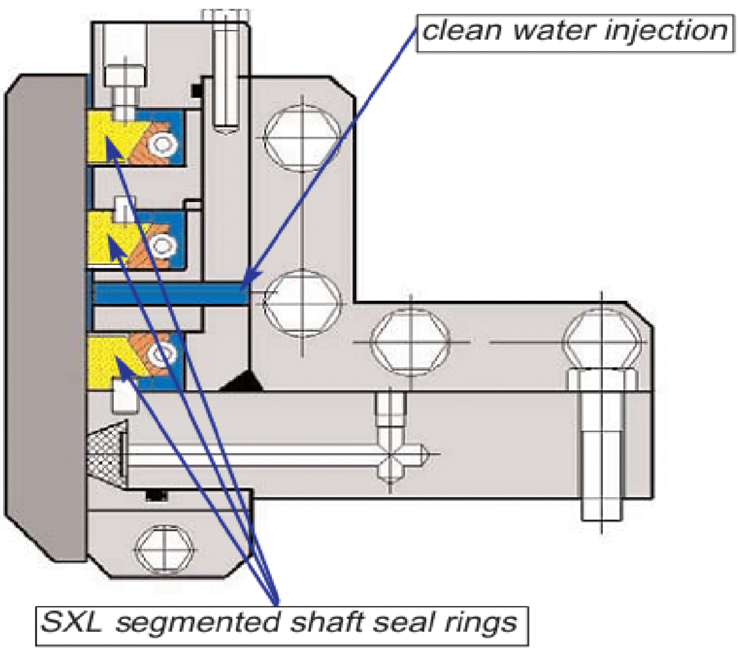

Figure (2.3): section of shaft seal.

In sennar hydropower station, the original shaft seals as show in figure (2.4), the system consist of pump sucked the clean water from gland sum tank and injected it into stage 3 of shaft seal to oppose the pressure of raw water from river. And from stage 2 there is a pipe went to pressure control valve from side and from other side there was a pipe came from the raw water before the shaft seal to make an equalization between the two pressures by opening and closing valve to maintain the pressure of clean water is constant and a bit higher than the raw water pressure, also there is a float valve to topping the gland sum by clean water from external source. And the excess water that leaked while the units were operating had to be pumped to turbine pit into the common drain and then discharge it to the downstream.

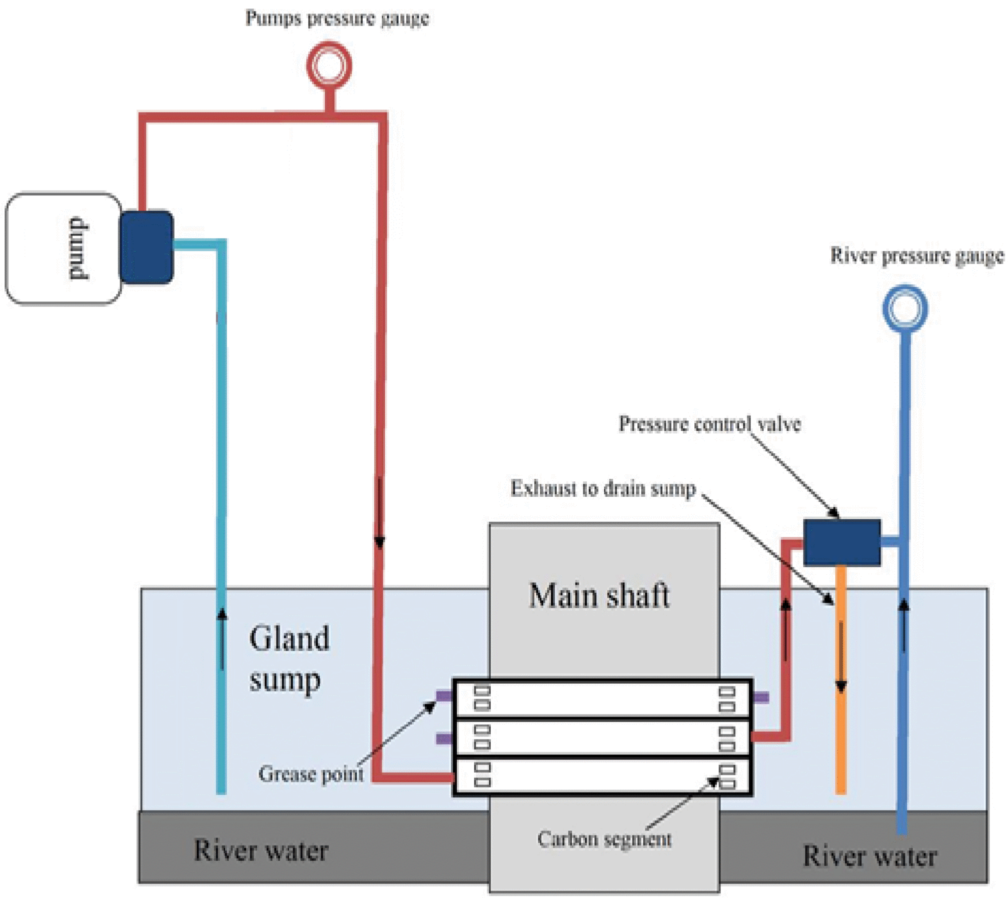

Figure (2.4): Clean water supply to gland system

2.2 Arduino uno:

Arduino can sense the environment by receiving input from a variety of sensors and can affect its surroundings by controlling lights, motors and other actuators. The microcontroller on the board is programmed using the Arduino Programming Language and the Arduino development environment. Arduino projects can be stand-alone or they can communicate with software running on a computer. Arduino is an open-source electronics platform based on easy-to-use hardware and software. It is a single-board microcontroller (microcontroller built onto a single printed circuit board). Arduino UNO is based on ATmega328. It has 14 digital input/output pins of which six can be used as PWM outputs. It has six analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller and is connected to the computer using USB cable or AC-DC adapter or battery to get started. [4], Figure [2.5] shows an Arduino UNO microcontroller board.

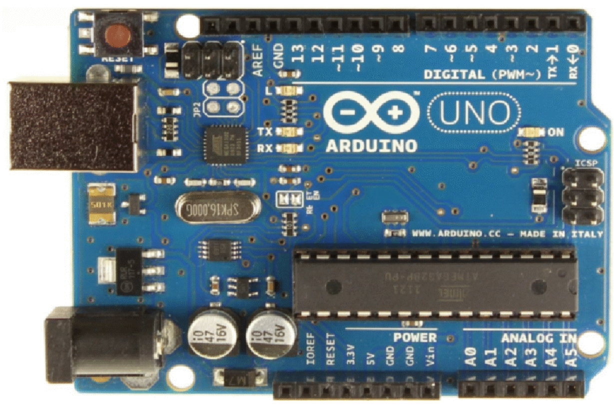

Figure (2.5): Arduino UNO microcontroller board

2.2.1 Summary of Arduino UNO:

To deal with arduino for any project you must know a lot of information about it, Table (2.1) shows the major information about arduino.

Table (2.1): Summary of Arduino UNO feature

Microcontroller	ATmega328
Operating voltage	5V
Input voltage	7V - 12V
Output voltage (limits)	6V – 12V
Digital I/O pins	16 (6 PWM pins)
Analog input pins	6
DC current per I/O pin	40 mA
DC current for 3.3V pin	50 mA
Flash Memory	32 KB
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

2.2.2 Power Supply of Arduino UNO:

The Arduino can be powered via USB connection or external power supply. Power source is selected automatically. External power can come from AC-DC supply or battery. The board can be operated on an external supply of 6V- 20V. The recommended voltage is 7V-12V. If the board is supplied with less than 7V, it will be unstable, and if it is supplied with more than 12V, the voltage regulator is overheated and may damage the board.

V_{IN}: The board is supplied with 5V through USB or from an external battery.

5V: This pin outputs a regulated 5V from the regulator onto the board. Power can be supplied with DC jack (7V – 12V), USB connector (5V), or the V_{IN} pin on the board.

3.3V: A 3.3V volt supply is generated by on-board regulator.

GND: Ground pin.

IOREF: Provides the voltage reference with which microcontroller operates. Selects the appropriate power source or enables voltage translators on the output working with 5V or 3.3V. [4]

2.2.3 Input and Output of Arduino UNO:

Each pin in the board can be used as input or output pin using pinMode( ), digitalWrite( ) and digitalRead( ) functions. Each pin is operated with 5V.

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- The UNO has six analog inputs, labeled A0 through A5, each of which provides 10 bits of resolution.
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library.
- AREF. Reference voltage for the analog inputs.

- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board. [4]

2.2.4 Communication of Arduino UNO:

Arduino UNO has several ways of communication with computers, other Arduinos or from other microcontrollers. ATmega328 provides UART TTL serial communication, which is available on digital pins 0 (RX) and 1 (TX). Arduino software has serial monitor which allows simple data to be sent to and from Arduino board. RX and TX LEDs will flash while data is being transferred via USB through the computer. [4]

2.2.5 Arduino UNO Programming and Reset:

Arduino UNO can be programmed with Arduino software. It comes with a boot loader that allows uploading the new code without any external hardware programmer. Arduino software uses Arduino C language which is similar to C++. Once the coding is over, it is uploaded to the board through USB. While uploading TX and RX, LEDs will flash. Choose the board as Arduino UNO and select serial port from serial port menu. This is likely to be COM3 or higher.

Arduino UNO has reset button in the board. When it is pressed, the board will reset.

Instead of doing so, UNO is designed in a way that allows it to reset by software running on a connected computer.

2.2.6 ATMega328:

Atmega328 is an 8-bit microcontroller with 32x8 general purpose-working registers. It has an on-chip 2-cycle multiplier. High-endurance non-volatile memory segments include 32K bytes of in-system self-programmable flash memory program, 1 K bytes of EEPROM and 2K bytes of internal SRAM. The write/erase cycles are set to 10,000 Flash/100,000 EEPROM.

The peripheral features of Atmega328 are:

- Two 8-bit timer/counters with separate prescaler and compare Mode
- One 16-bit timer/counter with separate prescaler, compare Mode, and capture mode
- Real-time counter with separate oscillator
- Six PWM Channels
- 8-channel 10-bit ADC in TQFP and QFN/MLF package
- 6-channel 10-bit ADC in PDIP Package
- Programmable Serial USART
- Master/Slave SPI Serial Interface
- Byte-oriented 2-wire Serial Interface (Philips I2C compatible)
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change

It has 23 programmable I/O lines. The operating voltage is in between 1.8V – 5.5 V and the temperature range is -400 C to 850 C. [5]

2.2.7 Arduino UNO Software:

Arduino UNO has its own software. It is similar to 'C' language. Code has to be written in the software provided and uploaded the code to the board which is connected to the computer using USB cable. When using development software, selection of Arduino UNO board and serial port is important as shown in Figure (2.6).

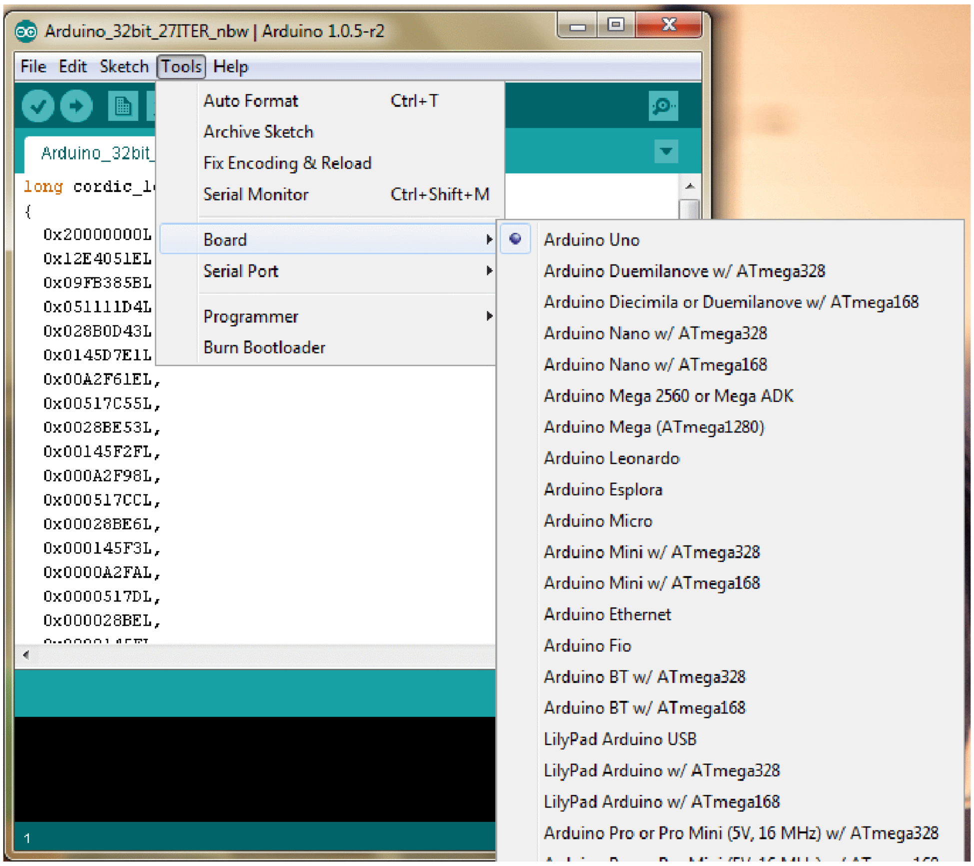

Figure (2.6): Selecting Arduino UNO board

2.3 PBSN Pressure Transmitter :

Figure (2.7) show Baumer's PBSN general purpose pressure transmitter series can now be used in media temperatures up to 150 °C. This is made possible by means of a cooling neck. The sensor, which is based on a thick film ceramic measuring cell, is also suitable for applications with aggressive media. This qualifies the transmitter for a wide variety of uses in fields as diverse as the petro-chemical, biotechnology and medical gas as well as food and beverage industries. [6]

Modularly constructed, the PBSN pressure transmitter series offers a flexible design to suit different customer needs, ensuring that their pressure measurement applications are served with an accurate, reliable and cost-effective product. Based on thick film technology, the measuring cells are suitable for gauge as well as for absolute pressure and cover a measuring range from -1…0 bar up to 0…600 bar. The standard accuracy is at 0.7%, or optionally 0.5%, of the full scale. [7]

Figure (2.7): Baumer's PBSN pressure transmitter

The thick film technology is based on a ceramic support material for the pressure measuring cell. This material is chemically neutral and compatible with most media, including very aggressive ones. These measuring cells are also characterised by excellent long-term stability and outstanding measuring performance in terms of linearity and hysteresis. [7]

A 4-20 mA output signal or voltage output signals are available and all common electrical connections and cable types are offered. The

PBSN can be delivered with different process connections made of stainless steel. Moreover, a wide range of gasket ring types are available. [7]

2.4 Ultrasonic Sensor Module:

2.4.1 Description of Ultrasonic Sensor :

The HC-SR04 ultrasonic sensor uses sonar to determine distance to an object like bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package.

From 2cm to 400cm or 1” to 13 feet. Its operation is not affected by sunlight or black material like sharp rangefinders are (although acoustically soft materials like cloth can be difficult to detect). It comes complete with ultrasonic transmitter and receiver module. As show in Figure (2.8). [7]

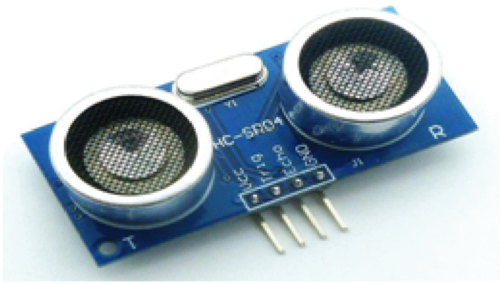

Figure (2.8): Ultrasonic Sensor Module

2.4.2 Features of Ultrasonic Sensor:

The feature of HC-SR04 ultrasonic sensor Module is:

- Power Supply :+5V DC
- Working Current: 15mA
- Effectual Angle: <15°

- Ranging Distance : 2cm – 400 cm/1″ – 13ft
- Resolution : 0.3 cm
- Measuring Angle: 30 degree
- Trigger Input Pulse width: 10uS

2.4.3 Ultrasonic Sensor Pin Configuration:

The table (2.2) below show the pin configuration of Ultrasonic Sensor module

Table (2.2): Ultrasonic Sensor Pin Configuration

Pin	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

2.4.4 Working of Ultrasonic Sensor:

As shown above the HC-SR04 Ultrasonic (US) sensor is a 4 pin module, whose pin names are Vcc, Trigger, Echo and Ground respectively. This sensor is a very popular sensor used in many applications where measuring distance or sensing objects are required.

The module has two eyes like projects in the front which forms the Ultrasonic transmitter and Receiver. The sensor works with the formula Distance equal speed multiplied by time

The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in air and when it gets objected by any material it gets reflected back toward the sensor this reflected wave is observed by the Ultrasonic receiver module as shown in the figure(2.9). [7]

Figure (2.9): Ultrasonic Sensor Module working

Now, to calculate the distance using the above formulae, we should know the Speed and time. Since we are using the Ultrasonic wave we know the universal speed of US wave at room conditions which is 330m/s. The circuitry inbuilt on the module will calculate the time taken for the US wave to come back and turns on the echo pin high for that same particular amount of time, this way we can also know the time taken. Now simply calculate the distance using a microcontroller or microprocessor. [7]

2.4.5 Using of Ultrasonic Sensor:

HC-SR04 distance sensor is commonly used with both microcontroller and microprocessor platforms like Arduino, the following guide is universally since it has to be followed irrespective of the type of computational device used.

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger and the Echo pins are both I/O pins and hence they can be connected to I/O pins of the microcontroller. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor.

The amount of time during which the Echo pin stays high is measured by the microcontroller as it gives the information about the time taken for the wave to return back to the Sensor. Using this information the distance is measured as explained in the above heading. [7]

2.4.6 Applications of Ultrasonic Sensor:

- Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- Used to measure the distance within a wide range of 2cm to 400cm
- Can be used to map the objects surrounding the sensor by rotating it
- Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water. [7]

2.5 16x2 LCD Display Module:

An LCD is an electronic display module which uses liquid crystal to produce a visible image. The 16×2 LCD display is a very basic module commonly used in DIYs and circuits. The 16×2 translates o a display 16 characters per line in 2 such lines. In this LCD each character is displayed in a 5×7 pixel matrix.

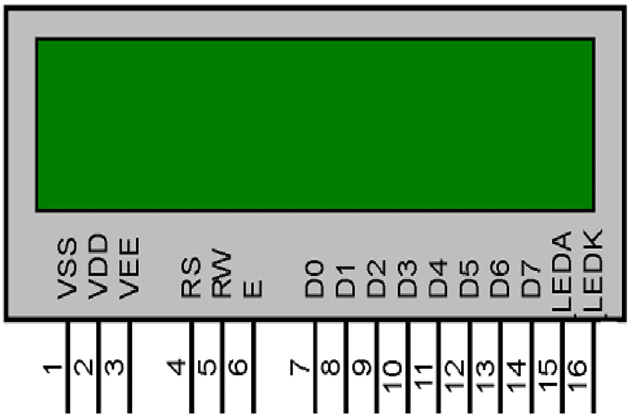

Figure (2.10): 16x2 LCD Display Module

Figure (2.10) show 16×2 LCD is named because it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. But the most used one is the 16*2 LCD. [8]

2.5.1 16X2 LCD pinout diagram:

The table (2.3) below show the pin diagram of LCD module that used to connect the LCD with the arduino.

Table (2.3): 16X2 LCD pinout Configuration

Pin No.	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; the best way is to use a variable resistor such as a potentiometer	Vo / VEE
4	Selects command register when low, and data register when high	RS (Register Select)
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given. Usually, we make it en=0 and when we want to execute the instruction we make it high en=1 for some milliseconds. After this we again make it ground that is, en=0.	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight VCC (5V)	Led+
16	Backlight Ground (0V)	Led-

2.5.2 16X2 LCD RS (Register select):

A 16X2 LCD has two registers, namely, command and data. The register select is used to switch from one register to other. RS=0 for command register, whereas RS=1 for data register. [8]

2.5.3 16X2 LCD Command Register:

The command register stores the command instructions given to the LCD. A command is an instruction given to LCD to do a predefined task like initializing it, clearing its screen, setting the cursor position, controlling display etc. Processing for commands happens in the command register. [8]

2.5.4 16X2 LCD Data Register:

The data register stores the data to be displayed on the LCD. The data is the ASCII value of the character to be displayed on the LCD. When we send data to LCD it goes to the data register and is processed there. When RS=1, data register is selected. [8]

2.5.5 Important command codes for LCD:

There are some preset commands instructions in LCD, which we need to send to LCD through some microcontroller. Some important command instructions are given below:

Table (2.4): Important command codes for LCD

Sr.No.	Hex Code	Command to LCD instruction Register
1	01	Clear display screen
2	02	Return home
3	04	Decrement cursor (shift cursor to left)
4	06	Increment cursor (shift cursor to right)
5	05	Shift display right
6	07	Shift display left
7	08	Display off, cursor off
8	0A	Display off, cursor on
9	0C	Display on, cursor off
10	0E	Display on, cursor blinking
11	0F	Display on, cursor blinking
12	10	Shift cursor position to left
13	14	Shift cursor position to right
14	18	Shift the entire display to the left
15	1C	Shift the entire display to the right
16	80	Force cursor to beginning (1st line)
17	C0	Force cursor to beginning (2nd line)
18	38	2 lines and 5×7 matrix

2.6 DC motor pump:

DC motors have been used as pump drive motors due to their variable speed control ability, especially at low speeds, simple control system, high starting torque and good transient response. Brushed, wound-field DC motors have formed the primary type of DC motor used in pump applications for many years. But permanent magnet (PMDC) and brushless DC motors have seen greater adoption rates, primarily due to their simple and compact design, high efficiency and power density, a wide range of frame sizes, and their need for less maintenance [9]. Figure (2.11) show the permanent magnet DC motor pump that used in the project.

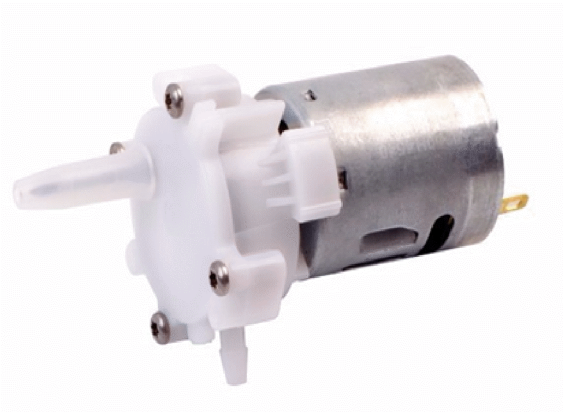

Figure (2.11): permanent magnet DC motor pump

2.7 Piezo buzzer:

The piezo buzzer produces sound based on reverse of the piezoelectric effect. The generation of pressure variation or strain by the application of electric potential across a piezoelectric material is the underlying principle. These buzzers show in figure (2.12) can be used alert a user of an event corresponding to a switching action, counter signal or sensor input. They are also used in alarm circuits.

Figure (2.12): Piezo buzzer

The buzzer produces a same noisy sound irrespective of the voltage variation applied to it. It consists of piezo crystals between two conductors. When a potential is applied across these crystals, they push on one conductor and pull on the other. This, push and pull action, results in a sound wave. Most buzzers produce sound in the range of 2 to 4 kHz. [10]

CHAPTER THREE

MODEL DESIGN

3.1 Original system overview:

In sennar hydropower station, the original shaft seals as discussed in chapter two, it consist of pump sucked the clean water from gland sum tank and injected it into shaft seal to opposed the pressure of raw water from river. And used pressure control to make an equalization between the two pressures by opening and closing valve to maintain the pressure of clean water is constant and a bit higher than the raw water pressure.

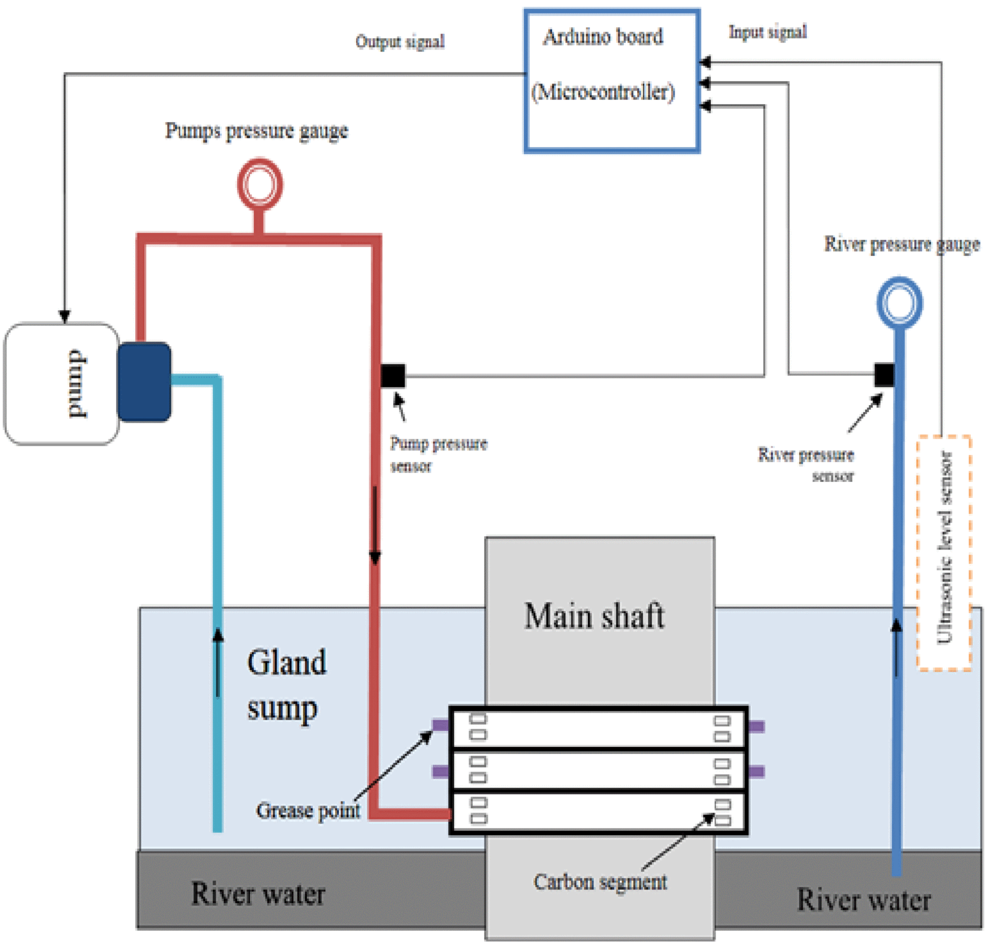

Figure (3.1): proposed modification of gland system

As show in figure (3:1) Arduino board use with two input pressure signals and one input level signal. Pressure signals are employed to read water pressure sensor before shaft seal (RWPS), and the other from pressure sensor of clean injected water (PPS), and the third input signal from ultrasonic distance sensor to read the level of water in gland sum. Arduino used to generate output signal to control the speed of the motor pump instead of pressure control valve, and maintained the pump pressure a bit higher than pressure of raw water. And other output signal that used as alarm in case of the level of water in gland sum was high.

Figure (3.2) and Figure (3.3) shows the electronic circuit diagram of development model, draw by fritzing and proteus programs.

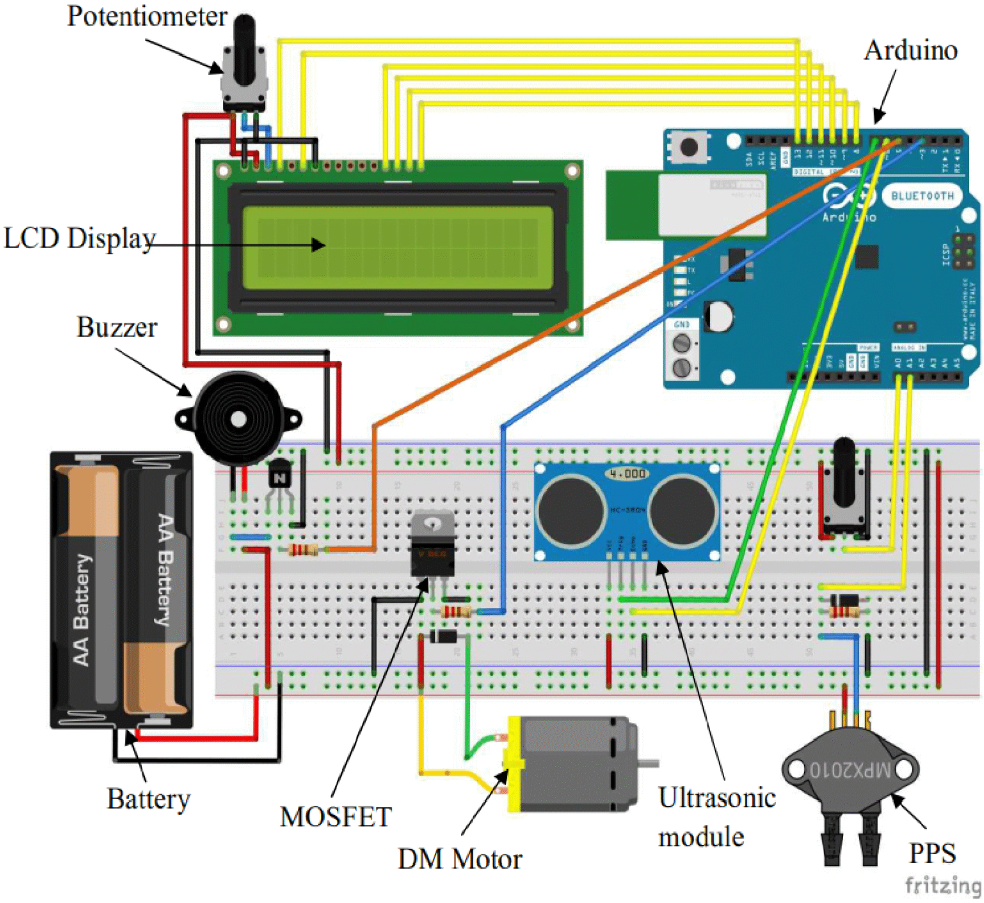

Figure (3.2): Complete circuit diagram of model (fritzing)

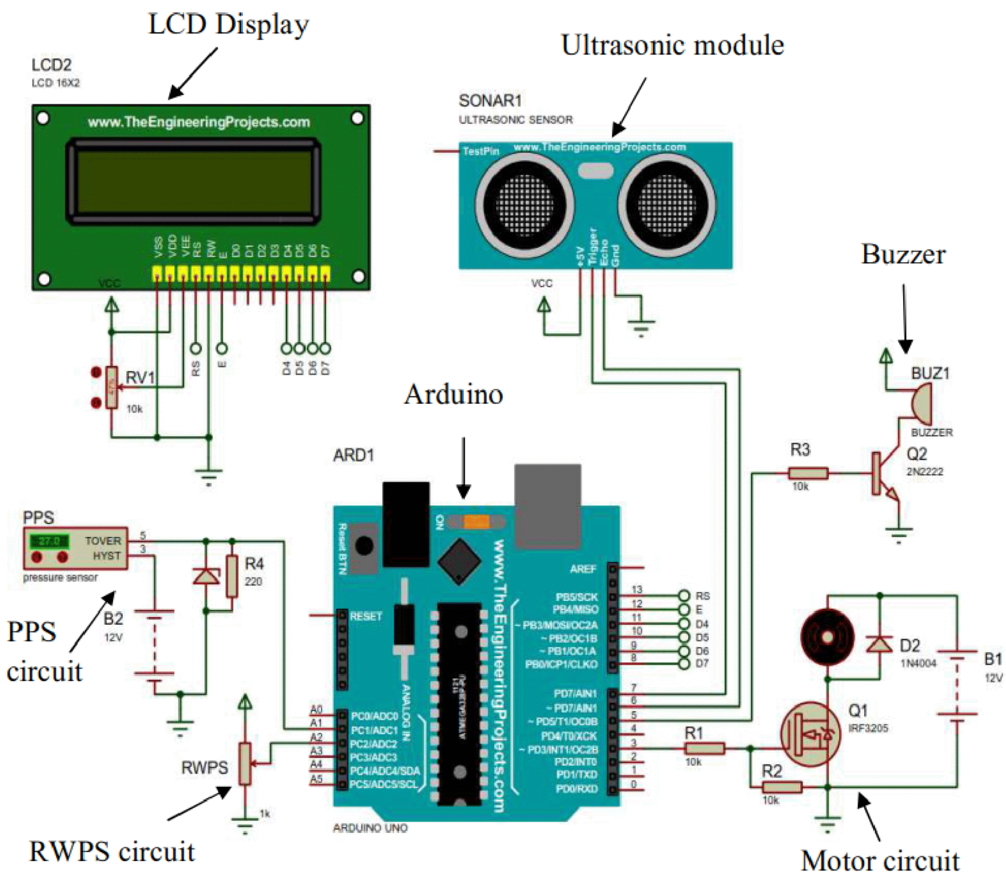

Figure (3.3): Complete circuit diagram of model (Proteus)

3.2 Design model and Hardware Implementation:

The major component of the model was small DC pump, used to simulate the real one, need to control the speed of motor to control the pressure of the system.

3.2.1 Speed Control of DC Motor pump Using PWM:

A DC motor (Direct Current motor) is the most common type of motor. DC motors normally have just two leads, one positive and one negative. If you connect these two leads directly to a battery, the motor will rotate. If you switch the leads, the motor will rotate in the opposite direction.

PWM control is a very commonly used method for controlling the power across loads. This method is very easy to implement and has high efficiency. PWM signal is essentially a high frequency square wave (typically greater than 1 KHz). The duty cycle of this square wave is varied in order to vary the power supplied to the load.

Duty cycle is usually stated in percentage and it can be expressed using the equation: $\% \text{ Duty cycle} = (T_{ON}/(T_{ON} + T_{OFF})) *100$.

Where T_{ON} is the time for which the square wave is high and T_{OFF} is the time for which the square wave is low. When duty cycle is increased the power dropped across the load increases and when duty cycle is reduced, power across the load decreases. The block diagram of a typical PWM power controller scheme is shown in figure (3.4) below.

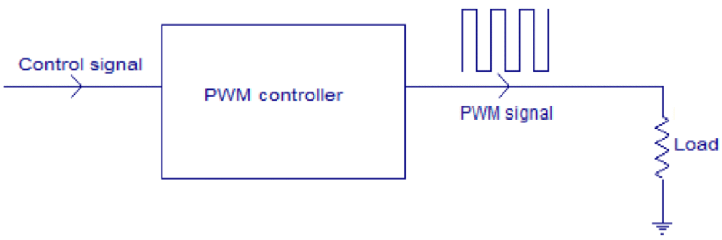

Figure (3.4): PWM power controller scheme

The control signal contains information on how much power has to be applied to the load. The PWM controller accepts the control signal and adjusts the duty cycle of the PWM signal according to the requirements. PWM waves with various duty cycles are shown in the figure (3.5) below.

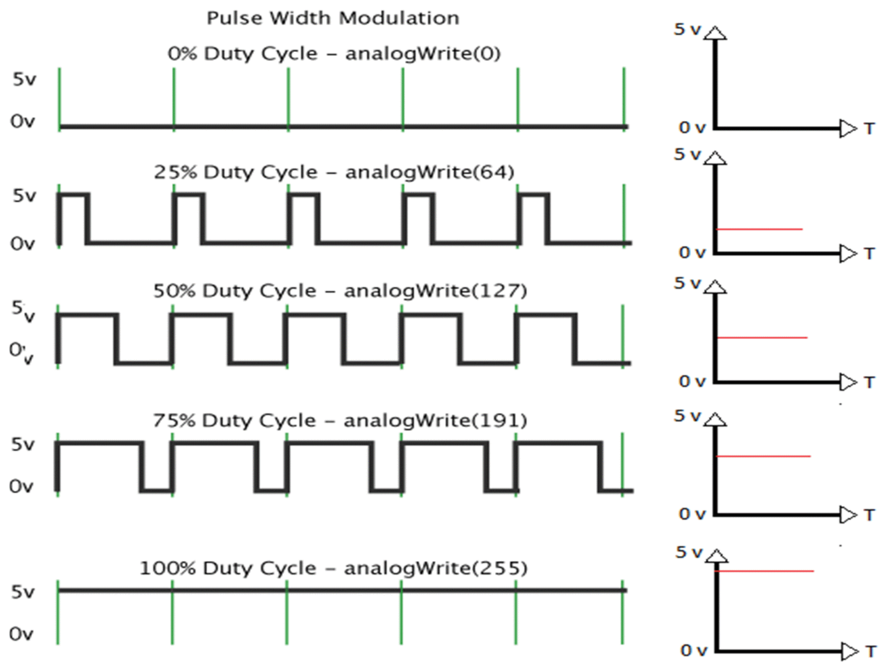

Figure (3.5): various duty cycles

In the above wave forms we can see that the frequency is same but ON time and OFF time are different.

The Arduino can only provide 40mA at 5V on its digital pins. Most DC motors require more current and/or voltage to operate. AMOSFT irf3205 transistor used to act as a digital switch, enabling the Arduino to control the DC motor pump.

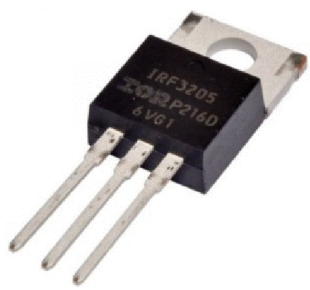

Figure (3.6): A MOSFT irf3205

When motor is spinning, suddenly turn off, a voltage spike was generating because the magnetic field inside it collapses. This can damage the transistor, to prevent this; use a diode which diverts the voltage spike around the transistor.

The PWM generated by arduino due to pin (3) to control switching of transistor, circuit diagram of pump speed control connected as figure (3.7):

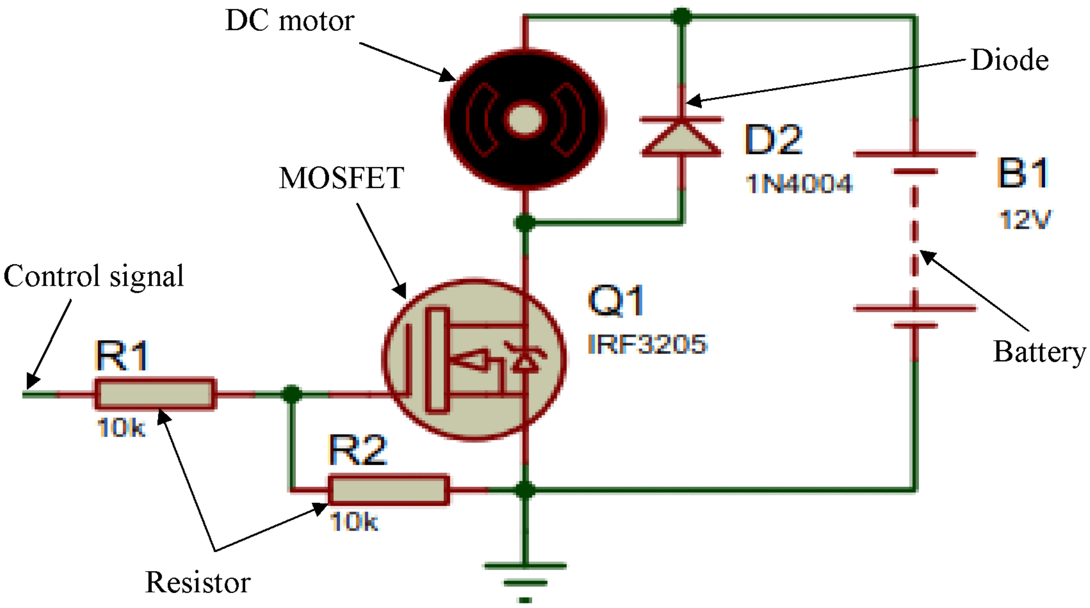

Figure (3.7): circuit diagram of pump speed control

3.2.2 Pressure sensor circuit diagram:

As mention above a Baumer's PBSN general purpose pressure transmitter is 4 to 20mA, the sensor generating the signal acts like a current source providing a constant current output signal for a given measurement independent of supply voltage or circuit impedance.

And arduino used voltage as analog input signal. The current can be converted to voltage output for measurement purposes at any point in the circuit by adding a load resistor in series. The voltage drop across the resistor will then vary proportionally with pressure and current. A 250 Ω resistor will produce a 1 to 5 volt output signal with 4-20mA current loop signal.

250Ω resistors aren't that common, so a more common value would be used instead - such as the commonly available 220Ω. Substituting the 250Ω for a 220Ω would give voltages:

@20mA: $V=IR = 0.02 \times 220 = 4.4V$

@04mA: $V=IR = 0.004 \times 220 = 0.88V$

Both comfortably within the Arduino's range and 5V zener diode use to prevent damage of arduino in the case that the resistor fails or becomes disconnected and get the full output voltage applied direct to the Arduino.

Figure (3.8) show the pressure transmitter circuit diagram arrangement that used in the project.

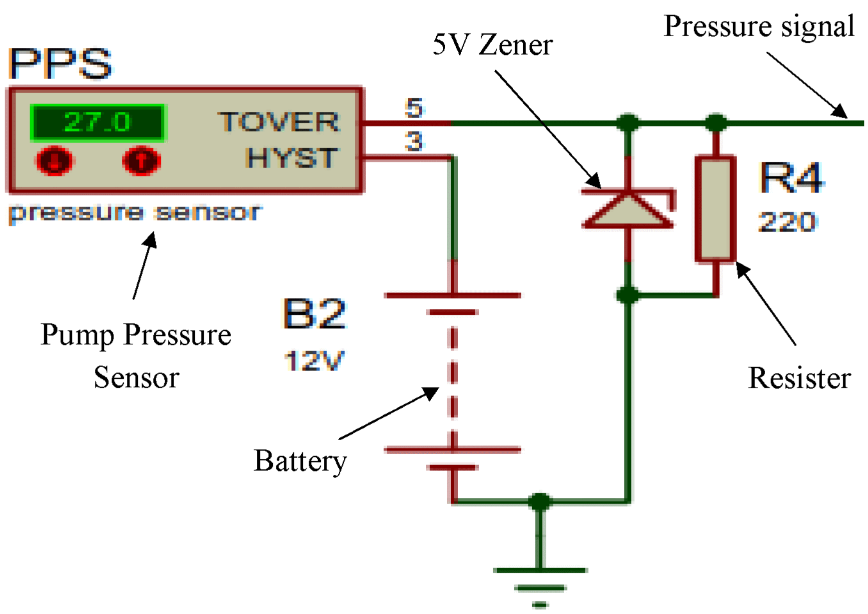

Figure (3.8): 4-20mA to 0.88-4.4V circuit diagram

When connecting up a 4-20mA transmitter to a circuit with a measurement load resistor it is important to ensure that there is sufficient supply voltage available across the transmitter positive and negative supply contacts particularly at full scale output.

The analog input signal comes from pump pressure sensor (PPS) connected to pin (A1), and for raw water pressure sensor (RWPS) I use potentiometer connected to pin (A2) to produce analog input signal instead of the sensor.

3.2.3 Ultrasonic Sensor Module circuit diagram:

Power the Sensor using a regulated +5V through the Vcc ad Ground pins of the sensor. The current consumed by the sensor is less than 15mA and hence can be directly powered by the on board 5V pins (If available). The Trigger pins connected to pins (7) of the arduino and the Echo pins connected to pins (6) of the arduino. To start the measurement, the trigger pin has to be made high for 10uS and then turned off. This action will trigger an ultrasonic wave at frequency of 40Hz from the transmitter and the receiver will wait for the wave to return. Once the wave is returned after it getting reflected by any object the Echo pin goes high for a particular amount of time which will be equal to the time taken for the wave to return back to the sensor. The amount of time during which the Echo pin stays high is measured by the microcontroller as it gives the information about the time taken for the wave to return back to the Sensor. Using this information to measure the level of water in gland sum and generate output signal as alarm.

3.2.4 Piezo buzzer circuit diagram:

piezo buzzer use as sound alarm when the level of water in gland sum get high. The buzzer connects from pin (5) of arduino through 2N2222 transistor as digital switch to protect the arduino.

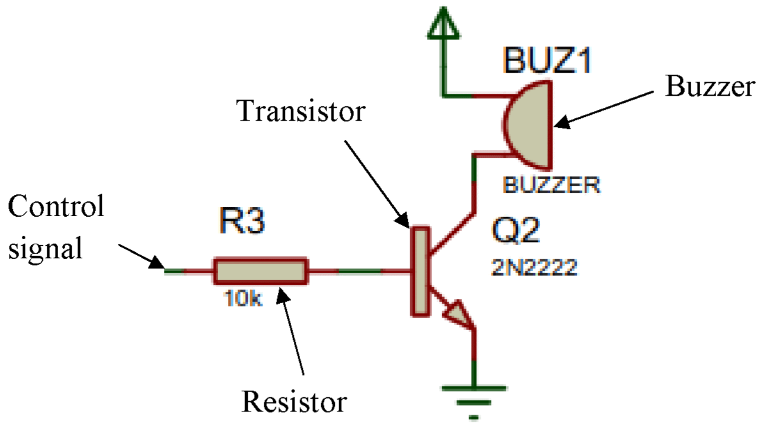

Figure (3.9): Piezo buzzer circuit diagram

3.2.5 LCD circuit diagram:

Potentiometer used to control the contrast of LCD. Other pins connected from pin (13) to pin (8)

3.3 System Hardware Implementation:

The model constructed by small 12V DC pump, sucked the water from small tank and delivered the water to same tank through a 2.5bar pressure transmitter which connected to pin A1as PPS input through current to voltage driver, a potentiometer connected to pin A2 as RWPS input, LCD connected from pin 13 to pin8 of arduino, DC motor connected to PWM pin 3 through MOSFET driver as output, also buzzer connected to pin 5 through 2N2222 transistor as output, trigger and echo of ultrasonic level sensor connected to pin 7 and pin 6 respectively,12VDC adapter used to operate the pump and pressure transmitter as show in Figure (3.10) below.

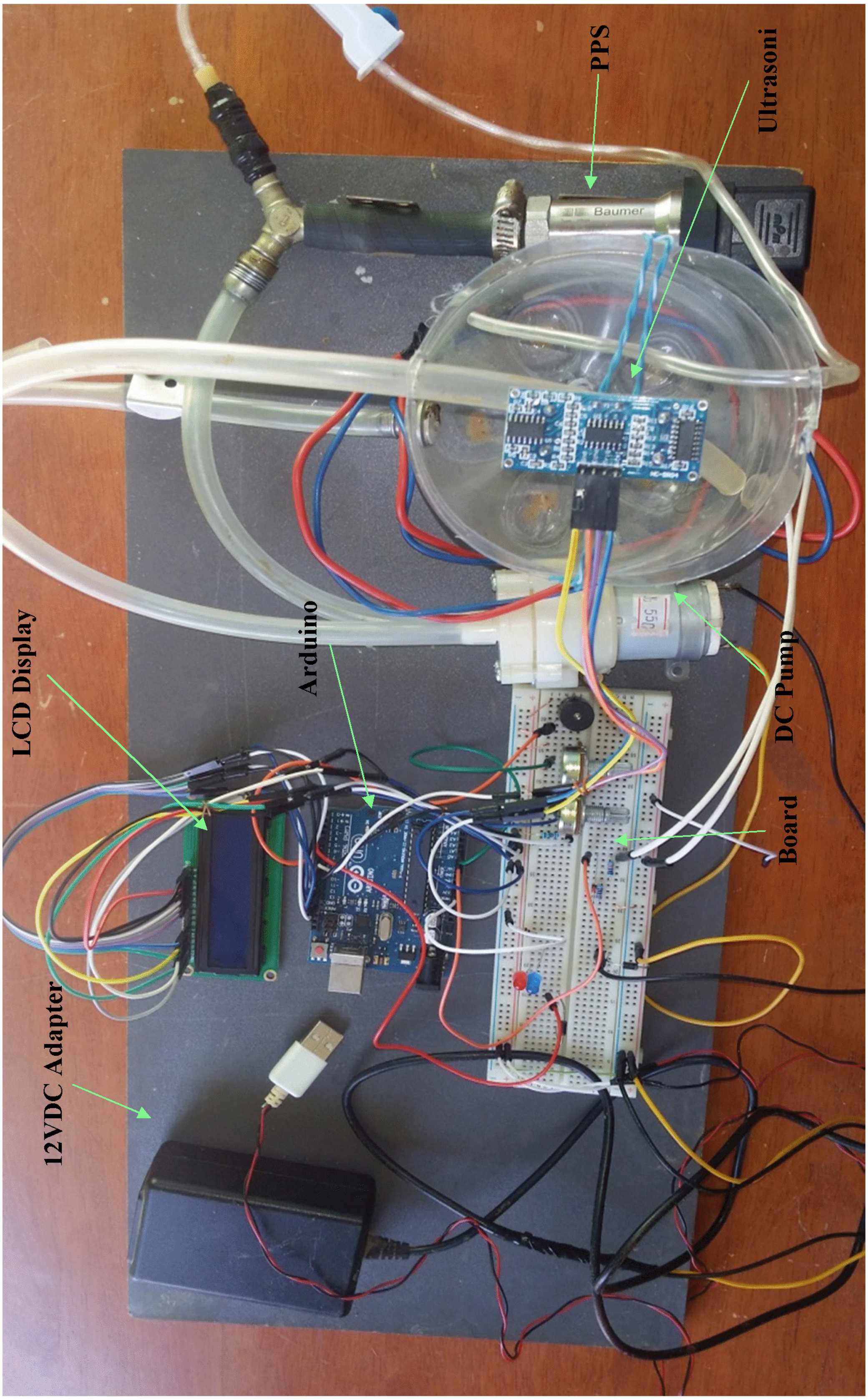

Figure (3.10): System Hardware.

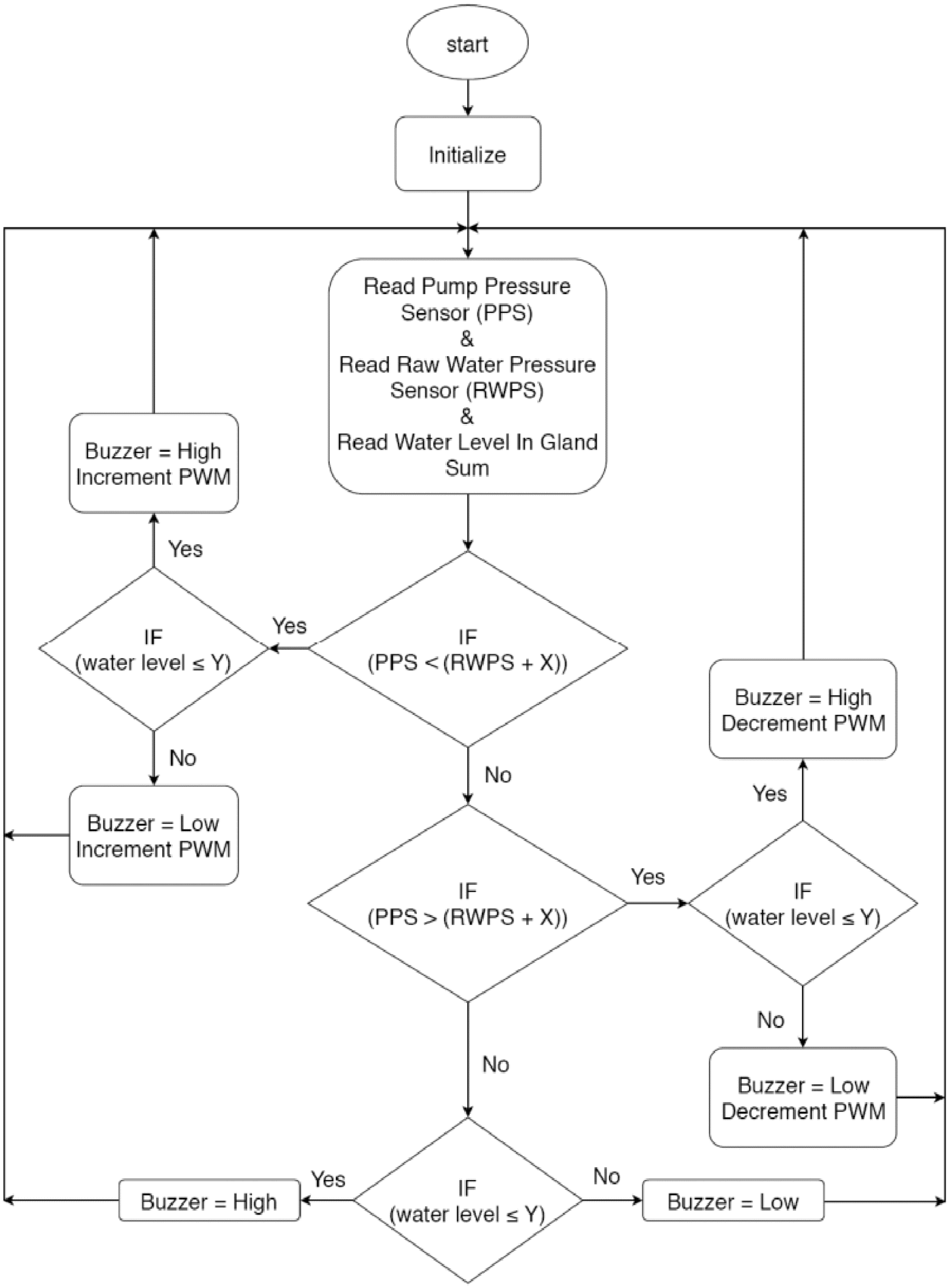

Figure (3.11): Flow chart of soft ware.

3.4 Operation Methods:

After starting, Arduino executed the program shown in appendix A, Arduino initializes the library with the numbers of the interface pins of LCD, define the pins that used in the system and define all variables.

Then setup routine (runs once when you press reset) such as initialize serial communication at 9600 bits per second, set up the LCD's number of columns and rows and declare the Pin as an INPUT and OUTPUT.

Then the loop routine (runs over and over again forever). Arduino following the steps shown in figure (3:11), so its read the value of water level in gland sum by ultrasonic module, read the value of raw water pressure sensor (RWPS) from potentiometer (use potentiometer instead of raw water pressure sensor) and read the value of pump pressure sensor (PPS) from Baumer’s PBSN pressure transmitter.

Make an average of 200 reading of the above sensor to insure that have an accurate value of reading.

In order to prevent abrasives particles entering from under the seal, the clean water supply must be at a bit higher pressure than the water pressure under the seal by (X kpa) value. And the water level in gland sum must not to be over (Y cm) value.

So, the program checks if:

- (PPS > (RWPS+X)) and (water level ≤ Y): the program will operate the buzzer and reduce the value of output PWM that use to control the speed of motor.
- (PPS > (RWPS+X)) and (water level > Y): the program will stop the buzzer and reduce the value of output PWM that use to control the speed of motor.
- (PPS < (RWPS+X)) and (water level ≤ Y): the program will operate the buzzer and increase the value of output PWM that use to control the speed of motor.
- (PPS < (RWPS+X)) and (water level > Y): the program will stop the buzzer and increase the value of output PWM that use to control the speed of motor.

- (PPS = (RWPS+X)) and (water level ≤ Y): the program will operate the buzzer and not change the value of output PWM that use to control the speed of motor.
- (PPS = (RWPS+X)) and (water level > Y): the program will stop the buzzer and not change the value of output PWM that use to control the speed of motor.

The program displays the value of PPS, RWPS, PWM value and water level in the LCD.

On site there is float valve that use to topping the gland sum if the water level gets lower than minimum.

Other potentiometer used for LCD resolution settings.

CHAPTER FOUR

RESULTS AND DISCUSSION

- In Chapter 3, physical hardware and electronic circuit of model of clean water supply to gland system were developed successfully.
- Several simulation cases were taken to test the circuit of the system. Two potentiometers were used instead of raw water pressure sensor (RWPS) and pump pressure sensor (PPS). Red led was used instead of buzzer alarm.
- Physical hardware model was operated and checked.
- Required operation condition:
    - If water level > 10cm → Buzzer get off
    - If water level ≤ 10cm → Buzzer get on
    - If PPS > (RWPS+10) → Reduce motor speed by reducing the value of the PWM
    - If PPS < (RWPS+10) → Increase motor speed by increasing the value of the PWM

4.1 Simulation cases:

Proteus 8 Professional used to draw and simulate the electronic circuit diagram; the arduino checked the above condition and generates PWM according to operation status. The simulation worked well and gave a good result, variable PWM from 10 to 255 was generated and motor speed from 57 rpm to 1000 rpm was controlled. Below the cases which applied to obtained the result.

4.1.1. Case one:

- Water level=10cm.
- PPS=80kpa & RWPS=71kpa → PPS < (RWPS+10)
- Simulation paused.

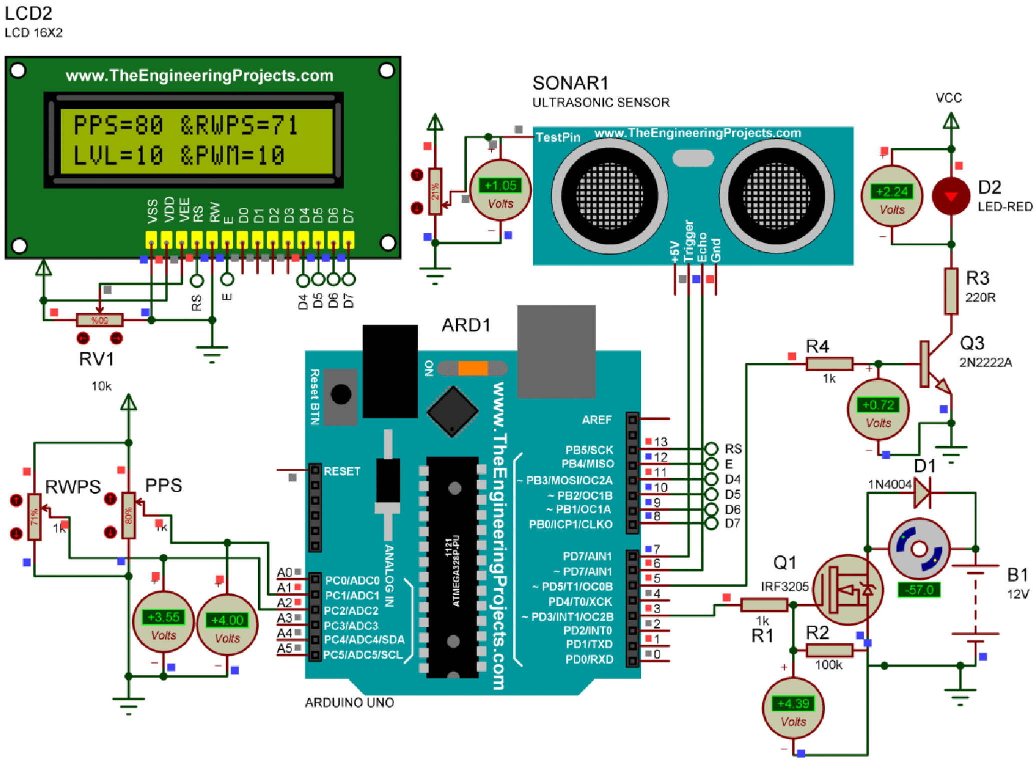

Figure (4.1): Simulation result for case one.

- Result:
    - ❖ As show in figure (4.1), Red led got on which mean the buzzer alarm was on because water level ≤10cm.
    - ❖ PWM reduced at minimum (10), so the speed of motor constant at minimum speed which is 57 rpm and would increased after simulation played.

4.1.2. Case two:

- Water level=11cm.
- PPS=80kpa & RWPS=71kpa → PPS < (RWPS+10)
- Simulation Paused approximately after 100ms from case one.

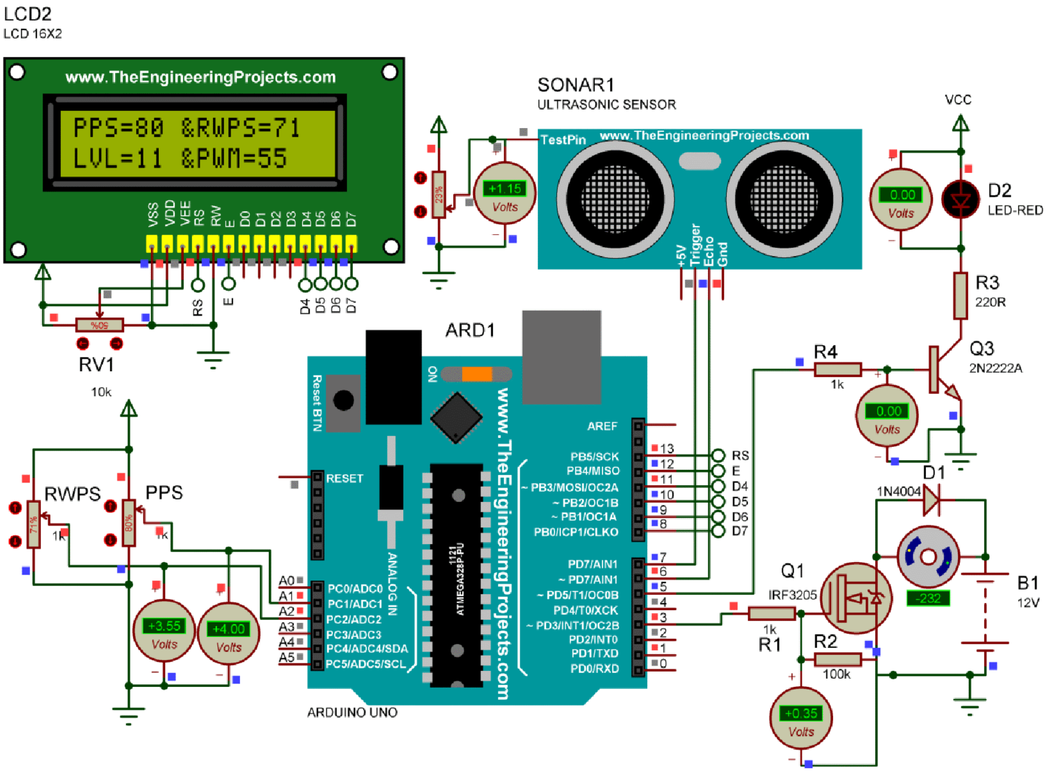

Figure (4.2): Simulation result for case two.

- **Result:**
  - ❖ As show in figure (4.2), Red led got off which mean the buzzer alarm was off because water level >10cm.
  - ❖ PWM increased to (55), so the speed of motor increased up to 232 rpm.

4.1.3. Case three:

- Water level=9cm.
- PPS=80kpa & RWPS=71kpa → PPS < (RWPS+10)
- Simulation Paused approximately after 100ms from case two.

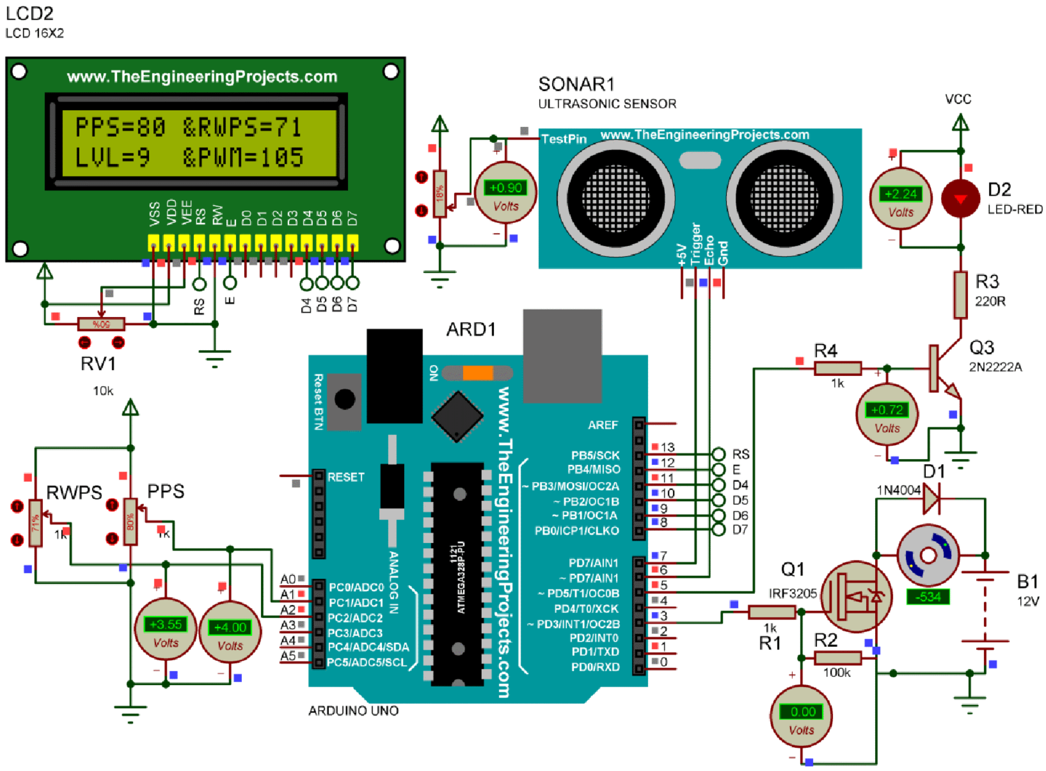

Figure (4.3): Simulation result for case three.

- **Result**:
  - ❖ As show in figure (4.3), Red led got on again which mean the buzzer alarm was on because water level ≤10cm.
  - ❖ PWM increased to (105), so the speed of motor increased up to 543 rpm.

4.1.4. Case four:

- Water level=12cm.
- PPS=80kpa & RWPS=71kpa → PPS < (RWPS+10)
- Simulation paused approximately after 100ms from case three.

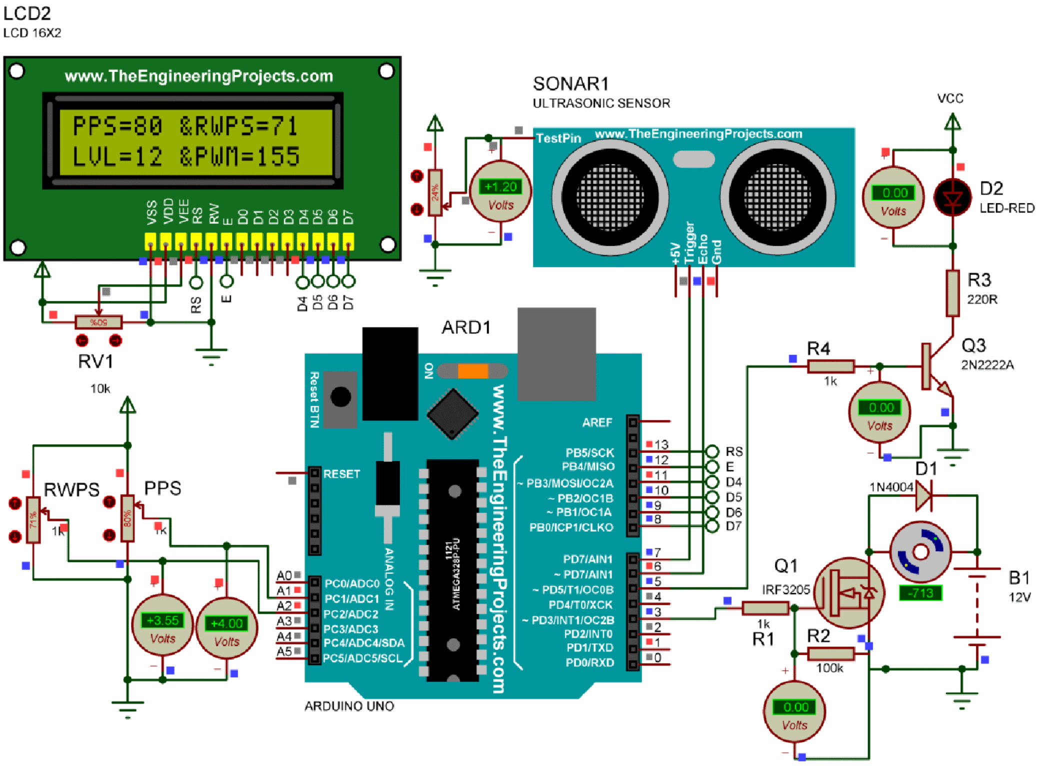

Figure (4.4): Simulation result for case four.

- **Result:**
  - ❖ As show in figure (4.4), Red led got off again which mean the buzzer alarm was off because water level >10cm.
  - ❖ PWM increased to (155), so the speed of motor increased up to 713rpm.

4.1.5. Case five:

- Water level=7cm.
- PPS=80kpa & RWPS=71kpa → PPS < (RWPS+10)
- Simulation paused approximately after 100ms from case four.

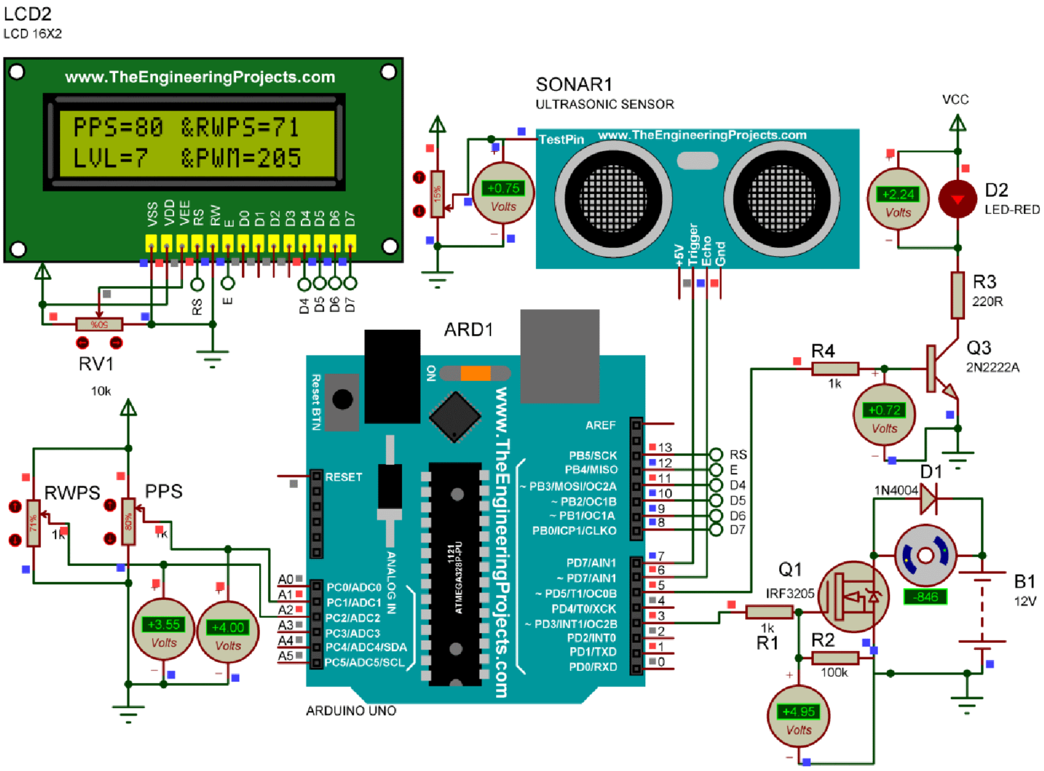

Figure (4.5): Simulation result for case five.

- **Result:**
  - ❖ As show in figure (4.5), Red led got on again which mean the buzzer alarm was on because water level ≤10cm.
  - ❖ PWM increased to (205), so the speed of motor increased up to 846 rpm.

4.1.6. Case six:

- Water level=14cm.
- PPS=80kpa & RWPS=71kpa → PPS < (RWPS+10)
- Simulation paused approximately after 100ms from case five.

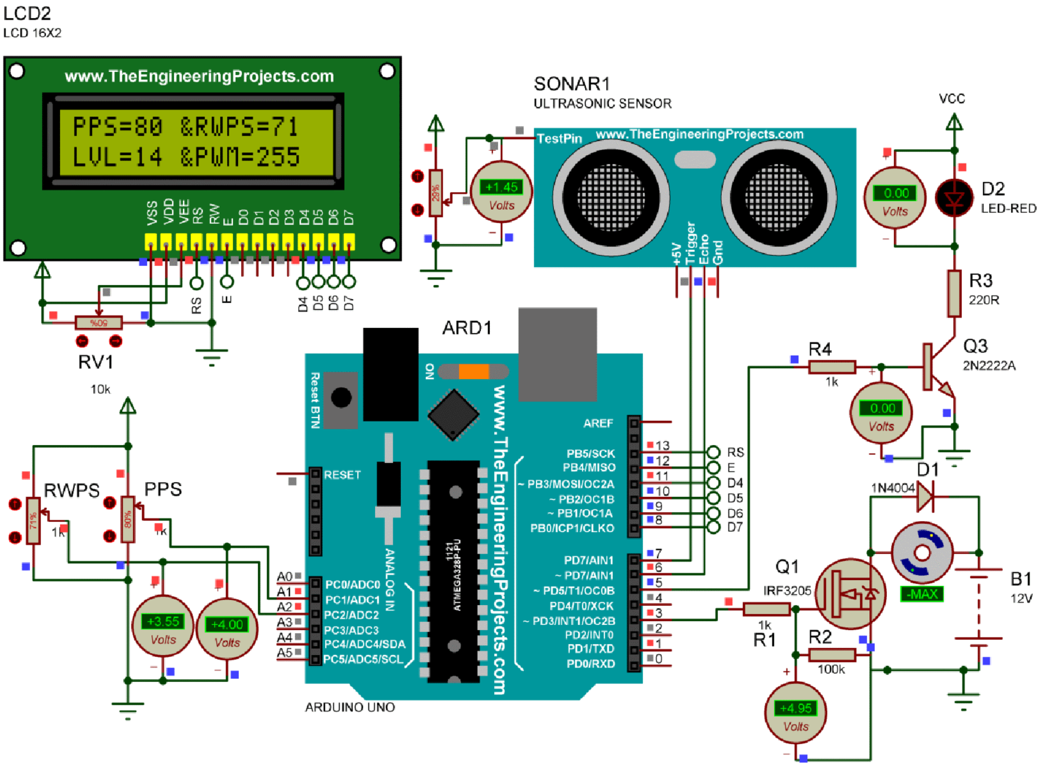

Figure (4.6): Simulation result for case six.

- **Result:**
  - ❖ As show in figure (4.6), Red led got off again which mean the buzzer alarm was off because water level >10cm.
  - ❖ PWM increased to (255), so the speed of motor increased up to 1000 rpm (max).

Below the cases when condition (PPS > (RWPS+10)) came true.

4.1.7. Case seven:

- Water level=10cm.
- PPS=80kpa & RWPS=69kpa → PPS > (RWPS+10)
- Simulation paused.

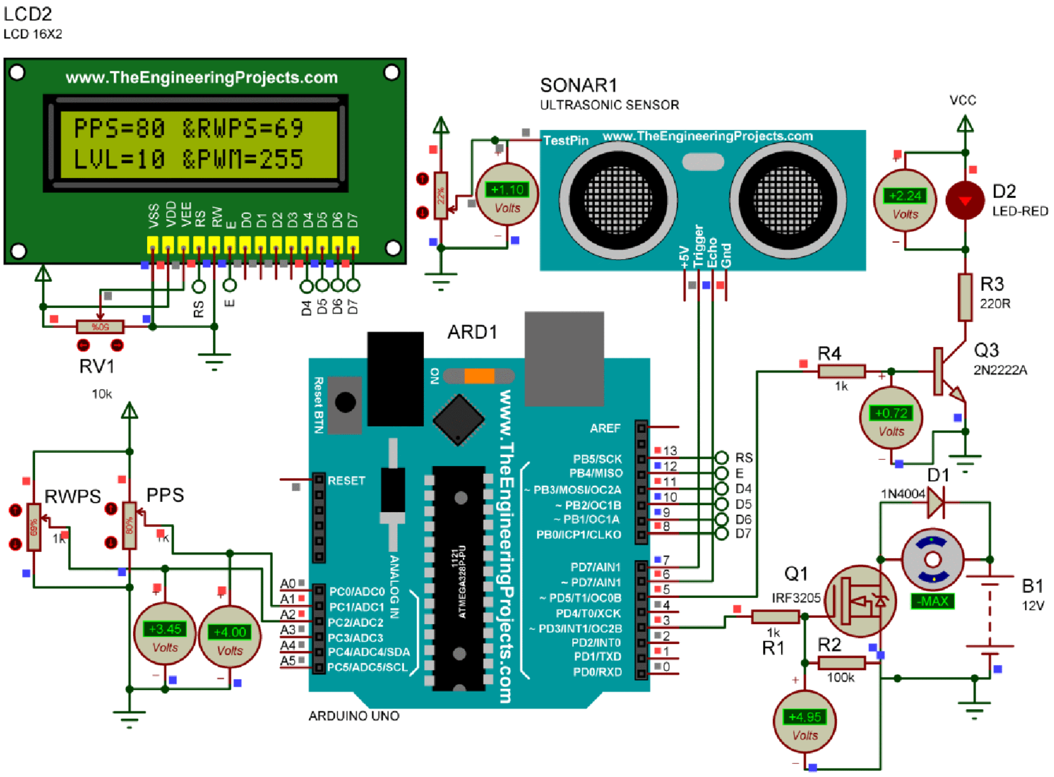

Figure (4.7): Simulation result for case seven.

- Result:
    - ❖ As show in figure (4.7), Red led got on which mean the buzzer alarm was on because water level ≤10cm.
    - ❖ PWM at maximum (255) from previous case, so the speed of motor constant at maximum speed which is 1000 rpm and would decreased after simulation played.

4.1.8. Case eight:

- Water level=11cm.
- PPS=80kpa & RWPS=69kpa → PPS > (RWPS+10)
- Simulation paused approximately after 100ms from case seven.

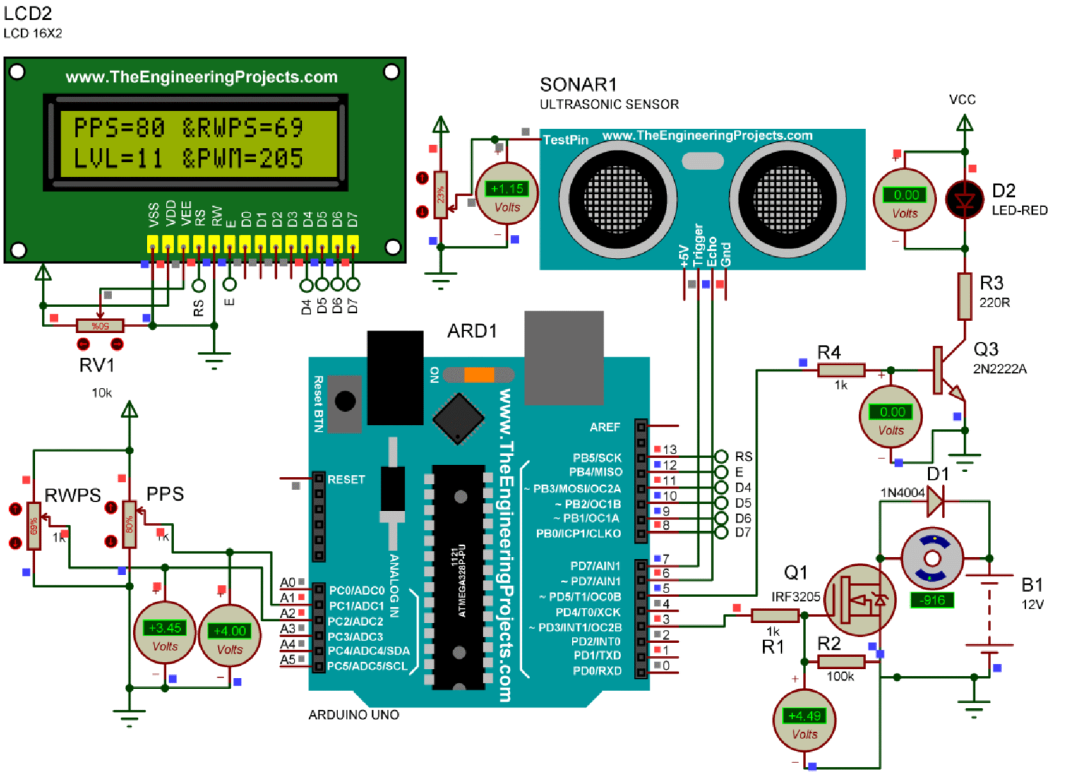

Figure (4.8): Simulation result for case eight.

- **Result:**
  - ❖ As show in figure (4.8), Red led got off which mean the buzzer alarm was off because water level >10cm.
  - ❖ PWM decreased to (205), so the speed of motor lowered to 916 rpm.

4.1.9. Case nine:

- Water level=9cm.
- PPS=80kpa & RWPS=69kpa → PPS > (RWPS+10)
- Simulation paused approximately after 100ms from case eight.

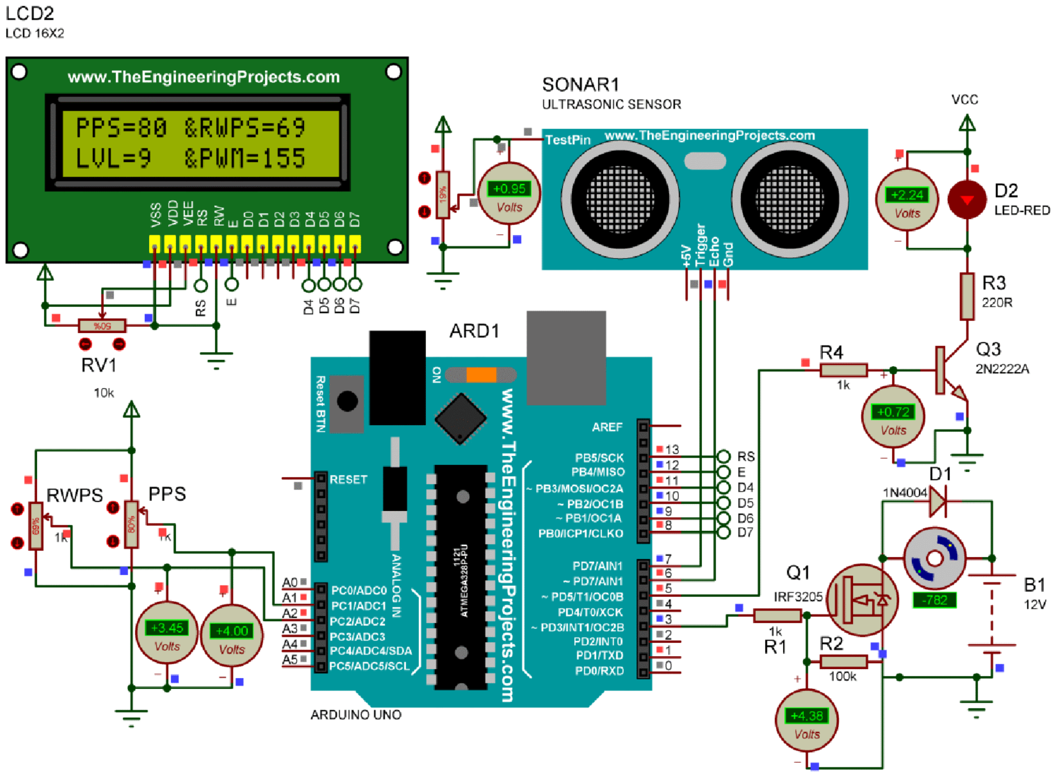

Figure (4.9): Simulation result for case nine.

- **Result**:
  - ❖ As show in figure (4.9), Red led got on again which mean the buzzer alarm was on because water level ≤10cm.
  - ❖ PWM Decreased to (155), so the speed of motor lowered to 782 rpm.

4.1.10. Case ten:

- Water level=12cm.
- PPS=80kpa & RWPS=69kpa → PPS > (RWPS+10)
- Simulation paused approximately after 100ms from case nine.

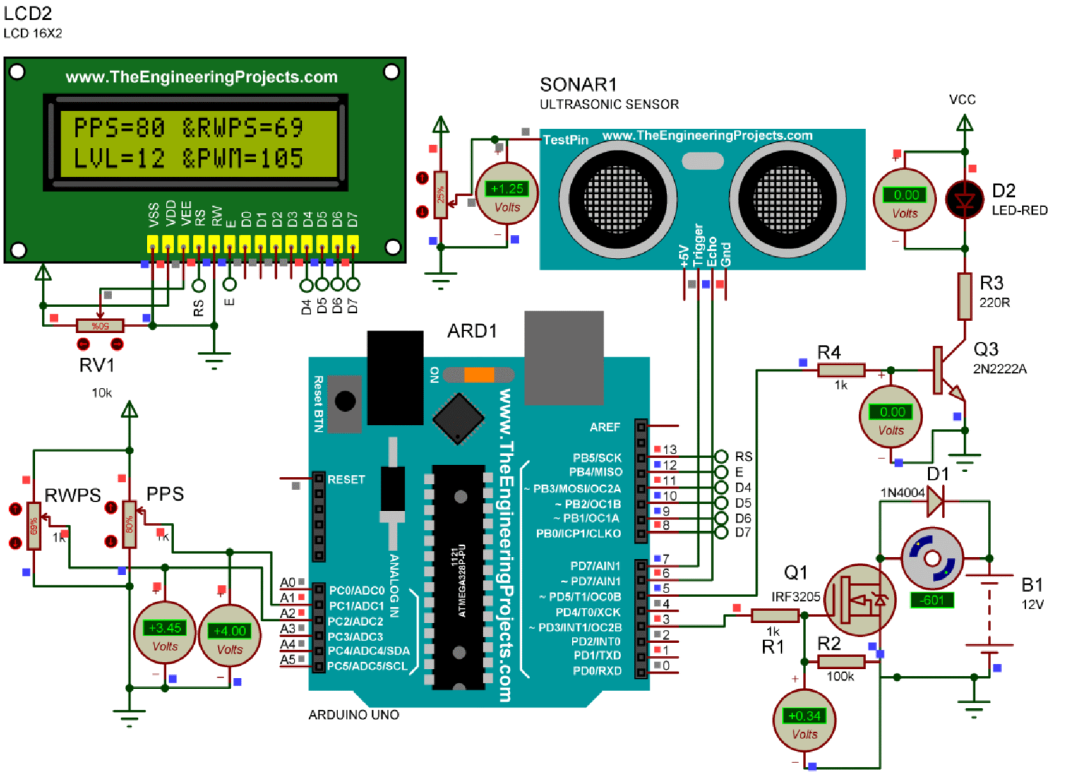

Figure (4.10): Simulation result for case ten.

- **Result:**
  - ❖ As show in figure (4.10), Red led got off again which mean the buzzer alarm was off because water level >10cm.
  - ❖ PWM Decreased to (105), so the speed of motor lowered to 601rpm.

4.1.11. Case eleven:

- Water level=7cm.
- PPS=80kpa & RWPS=69kpa → PPS > (RWPS+10)
- Simulation paused approximately after 100ms from case ten.

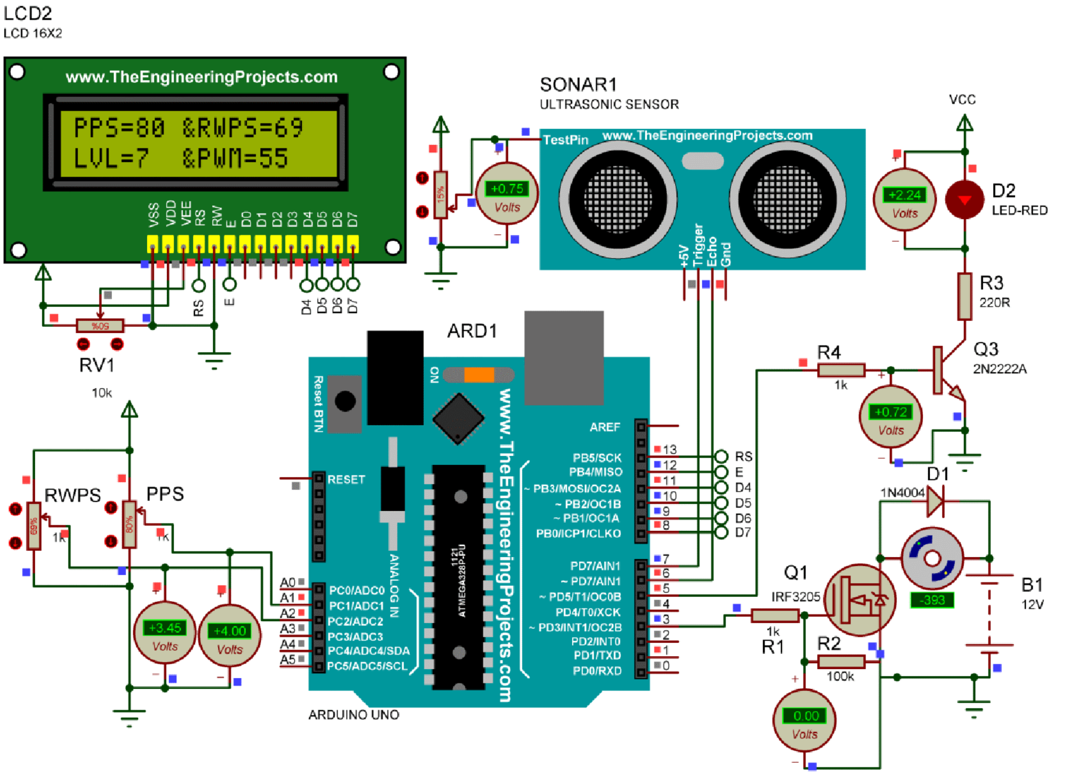

Figure (4.11): Simulation result for case eleven.

- **Result:**
  - ❖ As show in figure (4.11), Red led got on again which mean the buzzer alarm was on because water level ≤10cm.
  - ❖ PWM Decreased to (55), so the speed of motor lowered to 393 rpm.

4.1.12. Case twelve:

- Water level=14cm.
- PPS=80kpa & RWPS=69kpa → PPS > (RWPS+10)
- Simulation paused approximately after 100ms from case eleven.

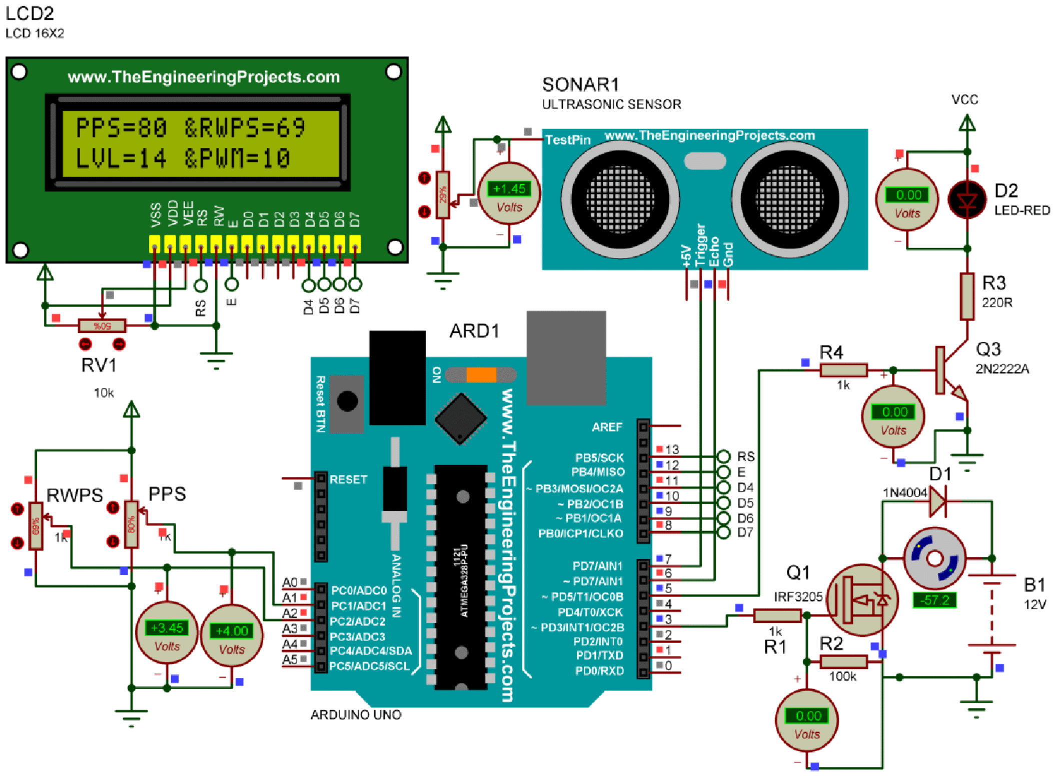

Figure (4.12): Simulation result for case twelve.

- **Result:**
  - ❖ As show in figure (4.12), Red led got off again which mean the buzzer alarm was off because water level >10cm.
  - ❖ PWM Decreased to (10), so the speed of motor lowered to 57.2 rpm.

figure (4:13) below shown all cases of simulation, clear that arduino fully controlled the speed of motor (RPM) by controlling the PWM according to condition of operation.

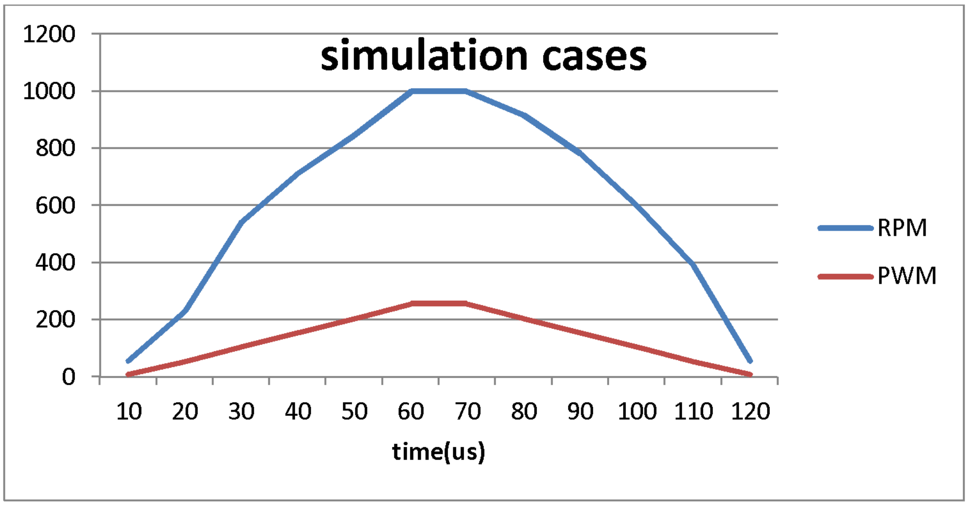

Figure (4.13): Simulation result for all cases.

4.2 Model test:

The required components of the model didn't available in the local market, so model was done by the existence component in the market.

After installing the hard-ware of the model and writing the required program code and tested the model, it worked well and obtained a good readings for the water level sensor, raw water pressure sensor (potentiometer) and pump pressure sensor, and gave a good operation for motor speed control to regulate the pressure of the pump.

Hardware model was tested in several condition of operation as shown below:

4.2.1. Water level sensor:

In all condition if the level of water dropped below 10cm from sensor → buzzer alarm was off

And when the level of water increased and became less than or equal 10cm → buzzer alarm was on

4.2.2. Motor speed control:

Initial value of pump pressure sensor (PPS) was set to 19kpa. And raw water pressure sensor (RWPS) was set to 0kpa. And PWM was set to 10.

1. When RWPS raised to 20kpa and condition (PPS<RWPS+10) was true: arduino started to increase PWM by 5 up to reach 100 and motor started to rotate till the pump pressure reached to 30kpa and then speed regulated by increasing or decreasing of PWM.
2. When RWPS raised to 30kpa and condition (PPS<RWPS+10) was true: arduino increased PWM by 5 up to reach 125 and motor increased the speed till the pump pressure reached to 40kpa and then speed regulated by increasing or decreasing of PWM.
3. When RWPS raised to 40kpa and condition (PPS<RWPS+10) was true: arduino increased PWM by 5 up to reach 155 and motor increased the speed till the pump pressure reached to 50kpa and then speed regulated by increasing or decreasing of PWM.
4. When RWPS raised to 50kpa and condition (PPS<RWPS+10) was true: arduino increased PWM by 5 up to reach 180 and motor increased the speed till the pump pressure reached to 60kpa and then speed regulated by increasing or decreasing of PWM.
5. When RWPS raised to 60kpa and condition (PPS<RWPS+10) was true: arduino increased PWM by 5 up to reach 210 and motor increased the speed till the pump pressure reached to

70kpa and then speed regulated by increasing or decreasing of PWM.

6. When RWPS raised to 70kpa and condition (PPS<RWPS+10) was true: arduino increased PWM by 5 up to reach 235 and motor increased the speed till the pump pressure reached to 80kpa and then speed regulated by increasing or decreasing of PWM.
7. When RWPS raised to 78kpa and condition (PPS<RWPS+10) was true: arduino increased PWM by 5 up to reach 255 (full speed) and motor rotated with full speed till the pump pressure reached to 88kpa and then speed regulated by decreasing of PWM. And then any increasing in RWPS didn't get to increase the PWM. Because of PWM get to maximum.

And vice versa when the RWPS decreased to any value the arduino decrement the PWM up to condition (PPS=RWPS+10) came true.

figure (4:14) below shown all cases of model test, clear that arduino fully controlled the speed of motor (PPS) by controlling the PWM according to condition of operation.

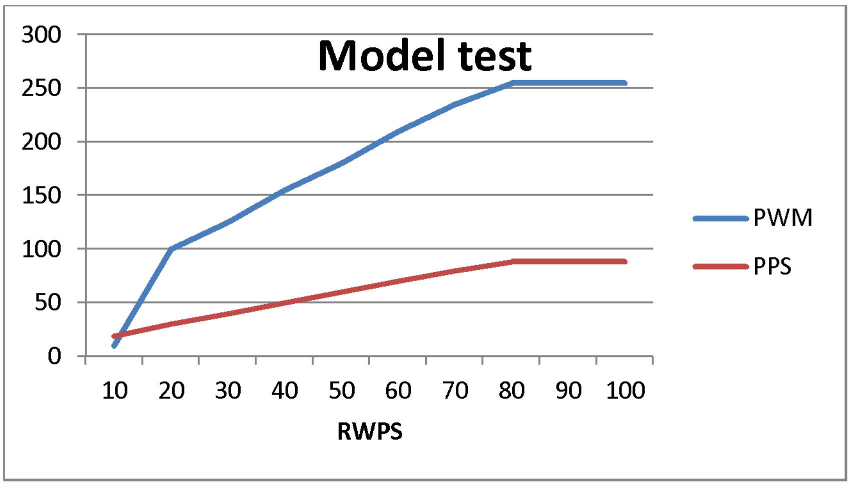

Figure (4.14): Model result for all cases.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Conclusion:

This thesis focuses on modeling of a clean water supply to gland system – in sennar power station. The following has been achieved in this research project:

- Design practical electronic circuit arduino base.
- Prepare Code program by using arduino software.
- Proteus 8 professional software is used to develop the simulation.

For Proteus simulation: as show in above simulation test cases, the buzzer gets on/off depended on the water level in the gland sum.

For motor speed control, as long as the condition (PPS < (RWPS+10)) was true, the arduino generated initial value of PWM =10 and then increased it by 5 up to 255 (full volts). And if the condition (PPS > (RWPS+10)) was true, the arduino reduced the previous value of PWM to reduced the speed of motor. And if the condition (PPS = (RWPS+10)) was true, the arduino wouldn't change value of PWM to kept speed of motor as the same value.

For hardware model: the buzzer worked perfectly, it's got on/off depended on the water level in the gland sum.

For motor speed control, the model worked perfectly and got satisfied results, and arduino regulate the pump pressure by controlling the speed of motor by generating the required PWM

5.2 Recommendations:

This project will be implemented in Sennar Hydro Power Plant as present modification. And before that it is recommended to develop a model used different simulation software, and reconstructed the model used real 2” DC pump instead of that one used in the model.

In the present study, Proportional control valve can be used to regulate the pressure instead of controlling the speed of DC motor pump, and the same PWM can use to control the opening percentage of proportional control valve.

And also alarm signal can use to open drainage solenoid valve to drain excess water of gland sun.

REFERENCES

1. Guojun (Gary) Ren, Ken Ogle, Hydro-Turbine Main Shaft Axial Sealsof Elastic Polymer–Principle and Practice, Thordon Bearings Inc. Canada, Feb 09, 2009

2. TURBINE AND GOVERNING SPECIFICATIONS, CHAPTER-7, Sample Technical Specifications of Large Hydraulic Turbines and Governing Equipment for Vertical Kaplan Turbines, IEC: 61366-2 (1998)

3. GRAPHITE COVA GMBH, BEARING AND SEALING COMPONENTS FOR MECHANICAL ENGINEERING, GRÜNTHAL 1 – 6 · D-90552 RÖTHENBACH/PEGNITZ, 2019

4. Arduino UNO "http://arduino.cc/en/Main/ArduinoBoardUNO"

5. Atmel Corporation, 8-bit Microcontroller with 4/8/16/32K Bytes In-SystemProgrammable Flash, 325 Orchard ParkwaySan Jose, CA 95131USA, 2009

6. Daniel Hunold, PBSNPressure transmitter for general industrial applications, baumer, Vendôme Cedex, France, 2019

7. Elijah J. Morgan, HCSR04 Ultrasonic Sensor, Nov. 16 2014

8. How 16×2 LCDs work | Build a basic 16×2 character LCD, (electronicsforu.com), December 4, 2018

9. Bhim Singh, Sanjeev Singh, State of the Art on Permanent Magnet Brushless DC Motor Drives, Indian Institute of Technology, Delhi, New Delhi, India, January 2009

10. Prof. S. Ben-Yaakov, Power Electronics of Piezoelectric Elements, June 2006

APPENDIX

APPENDIX (A): Arduino model code:

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
// with the arduino pin number it is connected to
const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

// Analog output pin that the Trigger  of Ultrasonic Sensor is attached to
const int ping_Pin = 7;
// Analog input pin that the Echo of Ultrasonic Sensor is attached to
const int echo_Pin = 6;
// Analog input pin that the buzzer is attached to
const int buzzer_Pin = 5;
// Analog input pin that the pump pressure sensor is attached to
const int pps_pin = A1;
// Analog input pin that the raw water pressure sensor is attached to
const int rwps_pin = A2;
// Analog output pin that the DC pump motor is attached to
const int pm_pin = 3;

float distance = 0;      // water level value
float times = 0;          // times for reflect ultrasonic sound
int pps = 0;             // value read from pump pressure sensor
int rwps = 0;             // value read from raw water pressure sensor
```

```
int pps_sum = 0;          // sum value read from pump pressure sensor
int rwps_sum = 0;         // sum value read from raw water pressure sensor
int distance_sum = 0;     //sum water level value
int pps_avg = 0;          // average value read from pump pressure sensor
int rwps_avg = 0;         // average value read from raw water pressure sensor
int distance_avg = 0;     //average water level value
int i = 0;                // counter
int x = 10;               // difference between pps and rwps
int y = 10;               // required gland sum water level
int PWM=10;

void setup() {
   // initialize serial communications at 9600 bps:
     Serial.begin(9600);
    // set up the LCD's number of columns and rows:
     lcd.begin(16, 2);
     pinMode(ping_Pin, OUTPUT);    // Trigger Pin of Ultrasonic Sensor as output
     pinMode(echo_Pin, INPUT);     // Echo Pin of Ultrasonic Sensor as input
     pinMode(buzzer_Pin, OUTPUT);  // buzzer as output
     pinMode(pps_pin, INPUT);      // pump pressure sensor as input
     pinMode(rwps_pin, INPUT);     // raw water pressure sensor as input
     pinMode(pm_pin, OUTPUT);      // DC pump motor as output
                 }
void loop() {
     delayMicroseconds(2);
     digitalWrite(ping_Pin, HIGH);
     delayMicroseconds(10);
```

```
    digitalWrite(ping_Pin, LOW);
    delayMicroseconds(2);
    times = pulseIn(echo_Pin, HIGH);   // stores the time span between the
transmitted and                                              //reflected waves
    distance = times * 0.0340 / 2;    //formula to calculate the distance in cm
    distance_sum =distance_sum + distance;


    pps = analogRead(pps_pin);        // read the analog pump pressure sensor:
    pps = map(pps, 0, 1023, 0, 255);  // map pump pressure sensor value :
    pps_sum = pps_sum + pps;


    rwps = analogRead(rwps_pin);      // read the analog rwa water pressure sensor:
    rwps = map(rwps, 0, 1023, 0, 255);     // map potentiometer value :
    rwps_sum =rwps_sum + rwps;


    if (i == 200){
      distance_avg = distance_sum/200;
      pps_avg = pps_sum /200;
      rwps_avg = rwps_sum/200;
       // Print a message to the LCD.
      lcd.clear();
      lcd.setCursor(0,0);    // set the cursor to (0,0):
      lcd.print("PPS=");
      lcd.print(pps_avg);
      lcd.setCursor(7,0);     // set the cursor to (7,0):
      lcd.print("&RWPS=");
      lcd.print(rwps_avg);
```

```
lcd.setCursor(0,1);     // set the cursor to (0,1):
lcd.print("LVL=");
lcd.print(distance_avg);
 lcd.setCursor(7,1);      // set the cursor to (7,0):
 lcd.print("&PWM=");
 lcd.print(PWM);
  if (distance_avg <= y){
                    digitalWrite(buzzer_Pin, HIGH);    //buzzer get on
                                }
  else{
                    digitalWrite(buzzer_Pin, LOW);     //buzzer get off
        }
 if ((PWM >= 10) && (PWM <= 255)){
          if (PWM > 10){
                    if ((pps_avg > rwps_avg + x)){
                    PWM--;
                     analogWrite(pm_pin, PWM);          // reduce the pump speed
:
                                                     }
                                 }
          if (PWM < 251){
                    if ((pps_avg < rwps_avg + x)){
                    PWM++;
                    analogWrite(pm_pin, PWM);          // incease the pump speed
:
                                                     }
                                 }
   distance_sum=0;
```

```
            rwps_sum=0;
            pps_sum=0;
            i=0;
        }
    }
    i++;
}
```